BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT/JP01/03414

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年 ４月２１日

出願番号
Application Number: 特願２０００－１８３７７０

出願人
Applicant (s): ソニー株式会社

２００１年 ３月 ２日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号 出証特２００１－３０１４４２９

【書類名】　特許願
【整理番号】　0000368703
【提出日】　平成12年 4月21日
【あて先】　特許庁長官殿
【国際特許分類】　H04N 5/76
【発明者】
【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
【氏名】　加藤　元樹
【発明者】
【住所又は居所】　東京都品川区北品川６丁目７番３５号　ソニー株式会社内
【氏名】　浜田　俊也
【特許出願人】
【識別番号】　000002185
【氏名又は名称】　ソニー株式会社
【代表者】　出井　伸之
【代理人】
【識別番号】　100082131
【弁理士】
【氏名又は名称】　稲本　義雄
【電話番号】　03-3369-6479
【手数料の表示】
【予納台帳番号】　032089
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1
【物件名】　要約書　1

【包括委任状番号】　9708842

【プルーフの要否】　　要

【書類名】　明細書

【発明の名称】　情報処理装置および方法、並びに記録媒体

【特許請求の範囲】

【請求項１】　AVストリームを記録媒体に記録する第１の記録手段と、

前記第１の記録手段により前記AVストリームの記録が開始された時点から終了される時点までに記録された前記AVストリームを１単位とし、前記単位毎に、前記AVストリームの属性情報として、前記AVストリームの記録モードに関する情報、前記AVストリームの記録レートの平均値に関する情報、前記AVストリームのうち符号化情報が同一である区間に関する情報、前記AVストリーム中のランダムアクセス可能な位置に関する情報、前記AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報を前記記録媒体に記録する第２の記録手段と

を含むことを特徴とする情報処理装置。

【請求項２】　前記記録媒体に２単位以上の前記AVストリームが記録されている状態において、ユーザが第１のAVストリームと第２のAVストリームを連続的に再生するように指示した場合、前記第１のAVストリームの所定の部分と前記第２のAVストリームの所定の部分から構成され、前記第１のAVストリームから前記第２のAVストリームに再生が切り換えられるとき再生される第３のAVストリームを作成する作成手段をさらに含み、

前記第１の記録手段は、前記AVストリームとして、前記作成手段により作成された前記第３のAVストリームを記録し、

前記第２の記録手段は、前記属性情報として、前記作成手段により作成された前記第３のAVストリームの属性情報を記録する

ことを特徴とする請求項１に記載の情報処理装置。

【請求項３】　AVストリームの記録を制御する第１の記録制御ステップと、

前記第１の記録制御ステップの処理で前記AVストリームの記録の制御が開始された時点から終了される時点までに記録が制御された前記AVストリームを１単位とし、前記単位毎に、対応するAVストリームの属性情報として、前記AVストリームの記録モードに関する情報、前記AVストリームの記録レートの平均値に関する情報、前記AVストリームのうち符号化情報が同一である区間に関する情報、前記

出証特２００１－３０１４４２９

AVストリーム中のランダムアクセス可能な位置に関する情報、前記AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報の記録を制御する第２の記録制御ステップと

を含むことを特徴とする情報処理方法。

【請求項４】　AVストリームの記録を制御する第１の記録制御ステップと、前記第１の記録制御ステップの処理で前記AVストリームの記録の制御が開始された時点から終了される時点までに記録が制御された前記AVストリームを１単位とし、前記単位毎に、対応するAVストリームの属性情報として、前記AVストリームの記録モードに関する情報、前記AVストリームの記録レートの平均値に関する情報、前記AVストリームのうち符号化情報が同一である区間に関する情報、前記AVストリーム中のランダムアクセス可能な位置に関する情報、前記AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報の記録を制御する第２の記録制御ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は情報処理装置および方法、並びに記録媒体に関し、特に、AVストリーム内のＩピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録する情報処理装置および方法、並びに記録媒体に関する。

【０００２】

【従来の技術】

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデ

ジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

【０００３】

ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG（Moving Picture Experts Group）２方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG１ Video、MPEG２ Video、DV方式をはじめとするデジタル記録方式であれば、１度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

【０００４】

しかしながら、このような記録方法は、供給されたビットストリームを１度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

【０００５】

また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、１度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

【０００６】

以上のように、デジタル放送が主流となる将来においては、データストリーマ

のように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

【０００７】

【発明が解決しようとする課題】

上述したような装置により、複数のデータ（例えば、映像データや音声データなどから構成される番組のデータ）が記録されている記録媒体を再生する際、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体化から、ＡＶストリームの読み出し位置の決定やストリームの復号といった処理を速やかに行わなくてはならないが、記録媒体に記録されるデータ量が増加するに従い、そのような処理を速やかにできないといった課題があった。

【０００８】

本発明はこのような状況に鑑みてなされたものであり、AVストリーム内のＩピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録することにより、AVストリームの読み出し位置の決定や復号処理を速やかに行えるようにすることを目的とする。

【０００９】

【課題を解決するための手段】

請求項１に記載の情報処理装置は、AVストリームを記録媒体に記録する第１の記録手段と、第１の記録手段によりAVストリームの記録が開始された時点から終了される時点までに記録されたAVストリームを１単位とし、単位毎に、AVストリームの属性情報として、AVストリームの記録モードに関する情報、AVストリームの記録レートの平均値に関する情報、AVストリームのうち符号化情報が同一である区間に関する情報、AVストリーム中のランダムアクセス可能な位置に関する情報、AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報を記録媒体に記録する第２の記録手段とを含むことを特徴とする。

【００１０】

記録媒体に２単位以上のAVストリームが記録されている状態において、ユーザが第１のAVストリームと第２のAVストリームを連続的に再生するように指示した

場合、第１のAVストリームの所定の部分と第２のAVストリームの所定の部分から構成され、第１のAVストリームから第２のAVストリームに再生が切り換えられるとき再生される第３のAVストリームを作成する作成手段をさらに含み、第１の記録手段は、AVストリームとして、作成手段により作成された第３のAVストリームを記録し、第２の記録手段は、属性情報として、作成手段により作成された第３のAVストリームの属性情報を記録することを特徴とする。

【００１１】

請求項３に記載の情報処理方法は、AVストリームの記録を制御する第１の記録制御ステップと、第１の記録制御ステップの処理でAVストリームの記録の制御が開始された時点から終了される時点までに記録が制御されたAVストリームを１単位とし、単位毎に、対応するAVストリームの属性情報として、AVストリームの記録モードに関する情報、AVストリームの記録レートの平均値に関する情報、AVストリームのうち符号化情報が同一である区間に関する情報、AVストリーム中のランダムアクセス可能な位置に関する情報、AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報の記録を制御する第２の記録制御ステップとを含むことを特徴とする。

【００１２】

請求項４に記載の記録媒体のプログラムは、AVストリームの記録を制御する第１の記録制御ステップと、第１の記録制御ステップの処理でAVストリームの記録の制御が開始された時点から終了される時点までに記録が制御されたAVストリームを１単位とし、単位毎に、対応するAVストリームの属性情報として、AVストリームの記録モードに関する情報、AVストリームの記録レートの平均値に関する情報、AVストリームのうち符号化情報が同一である区間に関する情報、AVストリーム中のランダムアクセス可能な位置に関する情報、AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報の記録を制御する第２の記録制御ステップとを含むことを特徴とする。

【００１３】

請求項１に記載の情報処理装置、請求項３に記載の情報処理方法、および、請求項４に記載の記録媒体においては、AVストリームの記録が開始された時点から

終了される時点までに記録されたAVストリームを１単位とし、単位毎に、AVストリームの属性情報として、AVストリームの記録モードに関する情報、AVストリームの記録レートの平均値に関する情報、AVストリームのうち符号化情報が同一である区間に関する情報、AVストリーム中のランダムアクセス可能な位置に関する情報、AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報が記録媒体に記録される。

【００１４】

【発明の実施の形態】

以下に、本発明の実施の形態について、図面を参照して説明する。図１は、本発明を適用した記録再生装置１の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置１は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【００１５】

端子１１には、アナログのビデオ信号が、端子１２には、アナログのオーディオ信号が、それぞれ入力される。端子１１に入力されたビデオ信号は、解析部１４とAVエンコーダ１５に、それぞれ出力される。端子１２に入力されたオーディオ信号は、AVエンコーダ１５に出力される。解析部１４は、入力されたビデオ信号からシーンチェンジなどの特徴点を抽出する。

【００１６】

AVエンコーダ１５は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ１６に出力する。

【００１７】

符号化ビデオストリームは、例えば、MPEG (Moving Picture Expert Group) ２方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG１方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化されたオーディオストリーム等である。マルチプレクサ１６は、入力されたビデオおよびオーディオのストリームを、入力システム情報

に基づいて多重化して、スイッチ１７を介して多重化ストリーム解析部１８とソースパケッタイザ１９に出力する。

【００１８】

多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ１９は、入力された多重化ストリームを、そのストリームを記録させる記録媒体１００のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC（誤り訂正）符号化部２０、変調部２１で所定の処理が施され、書き込み部２２に出力される。書き込み部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００にAVストリームファイルを書き込む（記録する）。

【００１９】

デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子１３に入力される。端子１３に入力されたトランスポートストリームの記録方式には、２通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子２４から制御部２３へ入力される。

【００２０】

入力トランスポートストリームをトランスペアレントに記録する場合、端子１３に入力されたトランスポートストリームは、スイッチ２５を介して多重化ストリーム解析部１８と、ソースパケッタイザ１９に出力される。これ以降の記録媒体１００へAVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【００２１】

入力トランスポートストリームを再エンコードした後に記録する場合、端子１３に入力されたトランスポートストリームは、スイッチ２５を介してデマルチプ

レクサ２６に入力される。デマルチプレクサ２６は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【００２２】

デマルチプレクサ２６により抽出されたストリーム（情報）のうち、ビデオストリームはAVデコーダ２７に、オーディオストリームとシステム情報はマルチプレクサ１６に、それぞれ出力される。AVデコーダ２７は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ１５に出力する。AVエンコーダ１５は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ１６に出力する。

【００２３】

一方、デマルチプレクサ２６から出力され、マルチプレクサ１６に入力されたオーディオストリームとシステム情報、および、AVエンコーダ１５から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部１８とソースパケットタイザ１９にスイッチ１７を介して出力される。これ以後の記録媒体１００へAVストリームが記録されるまでの処理は、上述の入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【００２４】

本実施の形態の記録再生装置１は、AVストリームのファイルを記録媒体１００に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部２３により作成される。制御部２３への入力情報は、解析部１４からの動画像の特徴情報、多重化ストリーム解析部１８からのAVストリームの特徴情報、および端子２４から入力されるユーザからの指示情報である。

【００２５】

解析部１４から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関係する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル（ＣＭ）の開始・終了点などの指定情報（マーク）であり、また

、その指定場所の画像のサムネイル画像の情報も含まれる。

【００２６】

多重化ストリーム解析部１８からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関係する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関係する情報（マーク）などである。

【００２７】

端子２４からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【００２８】

制御部２３は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、 AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベース、記録媒体１００の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部２０、変調部２１で処理されて、書き込み部２２へ入力される。書き込み部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００へデータベースファイルを記録する。

【００２９】

上述したアプリケーションデータベース情報についての詳細は後述する。

【００３０】

このようにして記録媒体１００に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生される場合、まず、制御部２３は、読み出し部２８に対して、記録媒体１００からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部２８は、記録媒体１００からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部２９、ECC復号部３０の処理

を経て、制御部２３へ入力される。

【００３１】

制御部２３は、アプリケーションデータベース情報に基づいて、記録媒体１００に記録されているPlayListの一覧を端子２４のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部２３へ入力される。制御部２３は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部２８に指示する。読み出し部２８は、その指示に従い、記録媒体１００から対応するAVストリームを読み出し復調部２９に出力する。復調部２９に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部３０の処理を経て、ソースデパケッタイザ３１出力される。

【００３２】

ソースデパケッタイザ３１は、記録媒体１００から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ２６に出力できるストリームに変換する。デマルチプレクサ２６は、制御部２３により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ２７に出力する。AVデコーダ２７は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子３２と端子３３から出力する。

【００３３】

また、ユーザインタフェースとしての端子２４から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部２３は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体１００からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部２８に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部２３は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部２８に指示する。

【００３４】

また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部２３は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部２８に指示する。

【００３５】

読み出し部２８は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【００３６】

次に、ユーザが、記録媒体１００に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体１００に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Ａという歌番組から歌手Ａの部分を再生し、その後続けて、番組Ｂという歌番組の歌手Ａの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子２４から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部２３に入力される。制御部２３は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成する。

【００３７】

ユーザが、記録媒体１００に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子２４から消去区間のイン点とアウト点の情報が制御部２３に入力される。制御部２３は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書き込み部２２に指示する。

【００３８】

ユーザが、記録媒体１００に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部２３は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エ

ンコードと再多重化を行う。

【００３９】

まず、端子２４から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部２３へ入力される。制御部２３は、読み出し部２８にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部２８は、記録媒体１００からデータを読み出し、そのデータは、復調部２９、ECC復号部３０、ソースデパケッタイザ３１を経て、デマルチプレクサ２６に出力される。

【００４０】

制御部２３は、デマルチプレクサ２６に入力されたデータを解析して、ビデオストリームの再エンコード方法（picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て）と、再多重化方式を決定し、その方式をAVエンコーダ１５とマルチプレクサ１６に供給する。

【００４１】

次に、デマルチプレクサ２６は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、「AVデコーダ２７に入力されるデータ」と「マルチプレクサ１６に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ２７で復号され、復号されたピクチャはAVエンコーダ１５で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ１６に入力される。

【００４２】

マルチプレクサ１６は、制御部２３から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部２０、変調部２１で処理されて、書き込み部２２に入力される。書き込み部２２は、制御部２３から供給される制御信号に基づいて、記録媒体１００にAVストリームを記録する。

【００４３】

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図２は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの２つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlayListの管理をする。ここでは、１つのAVストリームとその付属情報のペアを１つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【００４４】

１つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリーポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【００４５】

PlayListについて、図３を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。１つのPlayListは、Clipの中の再生区間の集まりである。所定のClipの中の１つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点（IN）とアウト点（OUT）の対で表される。従って、PlayListは、複数のPlayItemが集まることにより構成される。

【００４６】

PlayListには、２つのタイプがある。１つは、Real PlayListであり、もう１つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、それの参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real P

layListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【００４７】

Virtual PlayListは、Clipのデータを共有していない。従って、Virtual PlayListが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【００４８】

次に、Real PlayListの編集について説明する。図４（Ａ）は、Real PlayListのクリエイト(create：作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

【００４９】

図４（Ｂ）は、Real PlayListのディバイド(divide：分割)に関する図であり、Real PlayListが所望な点で分けられて、２つのReal PlayListに分割される操作である。この分割という操作は、例えば、１つのPlayListにより管理される１つのクリップ内に、２つの番組が管理されているような場合に、ユーザが１つ１つの番組として登録（記録）し直したいといったようなときに行われる。この操作により、Clipの内容が変更される（Clip自体が分割される）ことはない。

【００５０】

図４（Ｃ）は、Real PlayListのコンバイン(combine：結合)に関する図であり、２つのReal PlayListを結合して、１つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが２つの番組を１つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が１つにされる）ことはない。

【００５１】

図５（Ａ）は、Real PlayList全体のデリート(delete：削除)に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

【００５２】

図５（Ｂ）は、Real PlayListの部分的な削除に関する図であり、Real PlayLi

stの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【００５３】

図５（Ｃ）は、Real PlayListのミニマイズ(Minimize：最小化)に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayList にとって不必要なClipの、対応するストリーム部分は削除される。

【００５４】

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

【００５５】

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行、または、キャンセルする。または、Virtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

【００５６】

次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図６は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

【００５７】

図６（Ａ）に示したように、２つのReal PlayList１，２と、それぞれのReal PlayListに対応するClip１，２が存在している場合に、ユーザがReal PlayList１内の所定の区間（In１乃至Out１までの区間：PlayItem１）を再生区間として指示し、続けて再生する区間として、Real PlayList２内の所定の区間（In２乃至Out２までの区間：PlayItem２）を再生区間として指示したとき、図６（Ｂ）に示すように、PlayItem１とPlayItem２から構成される１つのVirtual PlayListが作成される。

【００５８】

次に、Virtual PlayList の再編集(Re-editing)について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入(insert)や追加(append)、Virtual PlayListの中のPlayItemの削除などがある。また、Virtual PlayListそのものを削除することもできる。

【００５９】

図７は、Virtual PlayListへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual PlayListへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【００６０】

Real PlayListとVirtual PlayListで共通の操作として、図８に示すようなPlayListの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList（図２０などを参照して後述する）によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【００６１】

次に、マーク（Mark）について説明する。マークは、ClipおよびPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば

、シーンチェンジ点などである。PlayListを再生する時、そのPlayListが参照するClipのマークを参照して、使用する事ができる。

【００６２】

PlayListに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。ClipまたはPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【００６３】

次にサムネイルについて説明する。サムネイルは、Volume、PlayList、およびClipに付加される静止画である。サムネイルには、２つの種類があり、１つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう１つは、マークが指しているシーンを表す画像である。

【００６４】

Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体１００、以下、記録媒体１００はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置１の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【００６５】

Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻０の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上２種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出

される必要がある。このため、すべてのメニューサムネイルを１つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図１０に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

【００６６】

一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。従って、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【００６７】

図１１は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図１２は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置１がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

【００６８】

また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは１つのファイルに格納するのがよい。Playlistはメニューサムネイル１つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【００６９】

図１３は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、およびClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内

容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

【００７０】

次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、２種類のCPIを用いる。１つは、EP_mapであり、もう一つは、TU_mapである。

【００７１】

EP_mapは、エントリーポイント(EP)データのリストであり、それはエレメンタリーストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリーポイントの場所を見つけるためのアドレス情報を持つ。１つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【００７２】

EP_mapは、主に２つの目的のために使用される。第１に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第２に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【００７３】

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される

【００７４】

本実施の形態では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【００７５】

SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置１が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【００７６】

デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体１００に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【００７７】

あるいは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【００７８】

例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトラン

スポートストリームは、共にISDBフォーマットに準拠しなければならない。

【００７９】

デジタル放送のストリームが、記録媒体１００に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP_mapが作成されてディスクに記録される。

【００８０】

または、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU_mapが作成されてディスクに記録される。

【００８１】

次にディレクトリとファイルについて説明する。以下、記録再生装置１をDVR（Digital Video Recording）と適宜記述する。図１４はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図１４に示したように、"DVR"ディレクトリを含むrootディレクトリ、"PLAYLIST"ディレクトリ、"CLIPINF"ディレクトリ、"M2TS"ディレクトリ、および"DATA"ディレクトリを含む"DVR"ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【００８２】

"DVR"ディレクトリの下には、 DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。"DVR"ディレクトリは、４個のディレクトリを含む。"PLAYLIST"ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが１つもなくても存在する。

【００８３】

"CLIPINF"ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが１つもなくても存在する。"M2TS"ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが

１つもなくても存在する。"DATA"ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【００８４】

"DVR"ディレクトリは、次に示すファイルをストアする。"info.dvr"ファイルは、 DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは１つのメニューサムネイルがなければならない。ファイル名は、memu.thmbに固定されるとする。メニューサムネイル画像が１つもない場合、このファイルは、存在しなくても良い。

【００８５】

"mark.thmb"ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは１つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が１つもない場合、このファイルは、存在しなくても良い。

【００８６】

"PLAYLIST"ディレクトリは、２種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。" xxxxx.rpls" ファイルは、１つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、１つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここで、"xxxxx"は、５個の０乃至９まで数字である。ファイル拡張子は、"rpls"でなければならないとする。

【００８７】

"yyyyy.vpls"ファイルは、１つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、１つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、５個の０乃至９まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

【００８８】

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、１つのファイルをストアする。"zzzzz.clpi" ファイルは、１つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

【００８９】

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

【００９０】

" DATA" ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【００９１】

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、" info.dvr" ディレクトリについて説明する。図１５は、" info.dvr" ディレクトリのシンタクスを示す図である。" info.dvr" ディレクトリは、３個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakerPrivateData()である。

【００９２】

図１５に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【００９３】

MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バ

イト数はゼロからカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。Ｎ１とＮ２は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【００９４】

DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図１６は、DVRVolume()のシンタクスを示す図である。図１６に示したDVR Volume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

【００９５】

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【００９６】

ResumeVolume()は、ボリュームの中で最後に再生したReal PlayListまたはVirtual PlayListのファイル名を記憶している。ただし、Real PlayListまたはVirtual PlayListの再生をユーザが中断した時の再生位置は、PlayListMark()において定義されるresume-markにストアされる。

【００９７】

図１７は、ResumeVolume()のシンタクスを示す図である。図１７に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが０にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

【００９８】

resume_PlayList_nameの１０バイトのフィールドは、リジュームされるべきReal PlayListまたはVirtual PlayListのファイル名を示す。

【００９９】

図１６に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolume は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをス

トアする。図１８は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの８ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

【０１００】

name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０１０１】

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが１にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事（再生される事）が許可される。このフラグが０にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【０１０２】

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが０にセットされているか、または、このフラグが１にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置１は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

【０１０３】

PINは、４個の０乃至９までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0

xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【０１０４】

次に図１５に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList(Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されているすべてのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

【０１０５】

図２０は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。

【０１０６】

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す３２ビットの符号なしの整数である。number_of_PlayListsの１６ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの１０バイトの数字は、PlayListのファイル名を示す。

【０１０７】

図２１は、TableOfPlayLists()のシンタクスを別実施の構成を示す図である。図２１に示したシンタクスは、図２０に示したシンタクスに、UIAppinfoPlayList（後述）を含ませた構成とされている。このように、UIAppinfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図２０に示したシンタクスを用いるとして

以下の説明をする。

【０１０８】

図１５に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置１のメーカが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカのプライベートデータを挿入できるように設けられている。各メーカのプライベートデータは、それを定義したメーカを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、１つ以上のmaker_IDを含んでも良い。

【０１０９】

所定のメーカが、プライベートデータを挿入したい時に、すでに他のメーカのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカのプライベートデータが、１つのMakersPrivateData()に含まれることが可能であるようにする。

【０１１０】

図２２は、MakersPrivateDataのシンタクスを示す図である。図２２に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバを示す４個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す３２ビットの符号なし整数を示す。

【０１１１】

mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエントリー数を与える１６ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカプライベートデータが２個以上存在してはならない。

【０１１２】

mpd_block_sizeは、１０２４バイトを単位として、１つのmpd_blockの大きさを与える１６ビットの符号なし整数である。例えば、mpd_block_size=１ならば、それは１つのmpd_blockの大きさが１０２４バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える１６ビットの符号なし整数である。maker_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

【０１１３】

maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコードを示す１６ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す１６ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

【０１１４】

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す３２ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd_blockは、同じサイズでなければならない。

【０１１５】

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図２３は、xxxxx.rpls（Real PlayList）、または、yyyyy.vpls（Virtual PlayList）のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、３個のオブジェクトから構成され、それらは、PlayList()、

PlayListMark()、およびMakerPrivateData()である。

【０１１６】

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１１７】

MakerPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１１８】

padding_word（パディングワード）は、PlayListファイルのシンタクスにしたがって挿入され、Ｎ１とＮ２は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【０１１９】

ここで、既に、簡便に説明したが、PlayListについてさらに説明する。ディスク内にあるすべてのReal PlayListによって、Bridge-Clip（後述）を除くすべてのClipの中の再生区間が参照されていなければならない。かつ、２つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【０１２０】

図２４を参照してさらに説明するに、図２４（Ａ）に示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図２４（Ｂ）に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしらのReal PlayListを参照することにより、必ず視聴することが可能である。

【０１２１】

図２４（Ｃ）に示したように、Virtual PlayListの再生区間は、Real PlayListの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在して

はならない。

【０１２２】

Real PlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、かつPlayList_typeが0（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、ひとつのSubPlayItemを含む事ができる。本実施の形態におけるPlayList()では、SubPlayIteはオーディオのアフレコの目的にだけに使用される、そして、１つのVirtual PlayListが持つSubPlayItemの数は、０または１でなければならない。

【０１２３】

次に、PlayListについて説明する。図２５は、PlayListのシンタクスを示す図である。図２５に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す３２ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す８ビットのフィールドであり、その一例を図２６に示す。

【０１２４】

CPI_typeは、１ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI_typeの値を示す。１つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す１６ビットのフィールドである。

【０１２５】

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、０から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す１６ビットのフィールドである。この値は、０または１である。付加的なオーディオストリームのパス(オーディオストリームパス）は、サブパスの一種で

ある。

【０１２６】

次に、図２５に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図２７は、UIAppInfoPlayListのシンタクスを示す図である。図２７に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、８ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示したテーブルに準拠する値に対応する。

【０１２７】

name_lengthは、８ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０１２８】

record_time_and_dateは、PlayListが記録された時の日時をストアする５６ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03 は、”0x20011223010203”と符号化される。

【０１２９】

durationは、PlayListの総再生時間を時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、01:45:30は、”0x014530”と符号化される。

【０１３０】

valid_periodは、PlayListが有効である期間を示す３２ビットのフィールドである。このフィールドは、８個の数字を４ビットのBinary Coded Decimal(BCD)

で符号化したものである。例えば、記録再生装置１は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、”0x20010507”と符号化される。

【０１３１】

maker_idは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置１）の製造者を示す１６ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す１６ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【０１３２】

playback_control_flagのフラグが１にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが０にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【０１３３】

write_protect_flagは、図２８（Ａ）にテーブルを示すように、１にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが０にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが１にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置１はユーザに再確認するようなメッセージを表示させる。

【０１３４】

write_protect_flagが０にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが１にセットされていても良い。ユーザが、Real PlayListを消去しようとする場合、記録再生装置１は、そのReal PlayListを消去する前に、上記Virtual PlayListの存在をユーザに警告するか、または、そのReal PlayListを”Minimize” する。

【０１３５】

is_played_flagは、図２８（Ｂ）に示すように、フラグが１にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、０にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

【０１３６】

archiveは、図２８（Ｃ）に示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す２ビットのフィールドである。ref_thumbnail_index のフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

【０１３７】

次にPlayItemについて説明する。１つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【０１３８】

PlayListが２つ以上のPlayItemから構成される時、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図２９に示す

【０１３９】

図３０は、PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time（図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time（図の中でOUT_time1と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

【０１４０】

現在のPlayItemのIN_time（図の中でIN_time2と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time（図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【０１４１】

図３１に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【０１４２】

PlayItemのシンタクスは、図３２に示すようになる。図３２に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【０１４３】

STC_sequence_idは、８ビットのフィールドであり、PlayItemが参照するSTC連

続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この８ビットフィールドは何も意味を持たず、０にセットされる。IN_timeは、３２ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図３３に示すように、PlayList()において定義されるCPI_typeによって異なる。

【０１４４】

OUT_timeは、３２ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図３４に示すように、PlayList()において定義されるCPI_typeによって異なる。

【０１４５】

Connection_Conditionは、図３５に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す２ビットのフィールドである。図３６は、図３５に示したConnection_Conditionの各状態について説明する図である。

【０１４６】

次に、BridgeSequenceInfoについて、図３７を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

【０１４７】

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

【０１４８】

図３７において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソース

パケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

【０１４９】

図３８は、BridgeSequenceinfoのシンタクスを示す図である。図３８に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

【０１５０】

RSPN_exit_from_previous_Clipの３２ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【０１５１】

RSPN_enter_to_current_Clipの３２ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【０１５２】

次に、SubPlayItemについて、図３９を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合だけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使

用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_information_file_ nameを含む。

【０１５３】

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【０１５４】

図４０は、SubPlayItemのシンタクスを示す図である。図４０に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【０１５５】

SubPath_typeの８ビットのフィールドは、sub pathのタイプを示す。ここでは、図４１に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【０１５６】

sync_PlayItem_idの８ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

【０１５７】

sync_start_PTS_of_PlayItemの３２ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlay

Item上のPTS(Presentaiotn Time Stamp)の上位３２ビットを示す。SubPath_IN_timeの３２ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する３３ビット長のPTSの上位３２ビットを示す。

【０１５８】

SubPath_OUT_timeの３２ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresenation_end_TSの値の上位３２ビットを示す。

Presentation_end_TS = PTS_out + AU_duration

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの９０kHz単位の表示期間である。

【０１５９】

次に、図２３に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図４２は、PlayListMarkのシンタクスを示す図である。図４２に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０１６０】

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す３２ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数である。number_of_PlayList_marks は、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図４３に示すテーブルに従って符号化される。

【０１６１】

mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図４４

に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する８ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

【０１６２】

character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示した値に対応する。name_lengthの８ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【０１６３】

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない事を示す。

【０１６４】

次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル) は、図４５に示すように６個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakerPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の"zzzzz"が使用される。

【０１６５】

図４５に示したzzzzz.clpi（Clip information fileファイル）のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１６６】

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【０１６７】

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData ()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word（パディングワード）は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。Ｎ１，Ｎ２，Ｎ３，Ｎ４、およびＮ５は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【０１６８】

次に、ClipInfoについて説明する。図４６は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル（Clip AVストリームまたはBridge-Clip AVストリームファイル）の属性情報をストアする。

【０１６９】

図４６に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す４個のキャラクター文字である。

version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す３２ビットの符号なし整数である。Clip_stream_typeの８ビットのフィールドは、図４７に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【０１７０】

offset_SPNの３２ビットのフィールドは、AVストリーム（Clip AVストリームまたはBridge-Clip AVストリーム）ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは０でなければならない。

【０１７１】

図４８に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例．RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【０１７２】

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

SPN_xxx = RSPN_xxx - offset_SPN

図４９に、SPN_xxx が、５である場合の例を示す。

【０１７３】

TS_recording_rateは、２４ビットの符号なし整数であり、この値は、DVRドライブ（書き込み部２２）へまたはDVRドライブ（読み出し部２８）からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clip

に対応するAVストリームが記録された時の日時をストアする５６ビットのフィールドであり、年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【０１７４】

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary Coded Decimal(BCD)で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【０１７５】

time_controlled_flag:のフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが１である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$TS\_average\_rate^{*}192/188^{*}(t - start\_time) - \alpha <= size\_clip(t) <= TS\_average\_rate^{*}192/188^{*}(t - start\_time) + \alpha$$

ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/second の単位で表したものである。

【０１７６】

また、上式において、ｔは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_clip(t)は、 時刻ｔにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻tまでに１０個のソースパケットが記録された場合、size_clip(t)は$10^{*}192$バイトである。αは、TS_average_rateに依存する定数である。

【０１７７】

time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録

する場合である。

【０１７８】

TS_average_rateは、time_controlled_flagが１にセットされている場合、この２４ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが０にセットされている場合、このフィールドは、何も意味を持たず、０にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【０１７９】

RSPN_arrival_time_discontinuityの３２ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

SPN_xxx = RSPN_xxx - offset_SPN

に基づいて算出される。

【０１８０】

reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが１である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが１である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが１である場合、transport_stream_IDのフィールドが有効であることを示す。is_servece_ID_validのフラグが１である場合、servece_IDのフィールドが有効であることを示す。

【０１８１】

is_ country_code_validのフラグが１である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration deascriotor（ISO/IEC13818-1で定義されている）が持つformat_identifierの値を示す。original_network_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【０１８２】

servece_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているservece_IDの値を示す。country_codeの２４ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の１６個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

【０１８３】

format_identifier、original_network_ID、transport_stream_ID、servece_ID,country_code 、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

【０１８４】

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値

によって特定される。図５０は、連続なＳＴＣ区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（ただし、後述するように、Clipの最大時間長は制限されている）。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【０１８５】

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図５１を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目（k>=0）のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【０１８６】

図５２は、STC_Infoのシンタクスを示す図である。図５２に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０１８７】

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【０１８８】

num_of_STC_sequencesの８ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。

所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、０から開始される。

【０１８９】

RSPN_STC_startの３２ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。

【０１９０】

次に、図４５に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図５３を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリーストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリーストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【０１９１】

program_sequenceは、同一の時刻において、ただ１つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ１つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【０１９２】

図５４は、ProgramInfoのシンタクスを示す図である。図５４に示したProgram Infoのシンタクを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０１９３】

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは１以上の値でなければならない。

【０１９４】

number_of_program_sequencesの８ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは１をセットされなければならない。RSPN_program_sequence_startの３２ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【０１９５】

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【０１９６】

PCR_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの８ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ

回数を示す。number_of_audiosの８ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【０１９７】

audio_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【０１９８】

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【０１９９】

図５５は、図５４に示したPrograminfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図５５に示したVideoCodingInfoのシンタクスを説明するに、video_formatの８ビットフィールドは、図５６に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【０２００】

frame_rateの８ビットフィールドは、図５７に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの８ビットフィールドは、図５８に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【０２０１】

図５９は、図５４に示したPrograminfoのシンタクス内のAudioCodingInfoのシ

ンタクスを示す図である。図５９に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図６０に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【０２０２】

audio_component_typeの８ビットフィールドは、図６１に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの８ビットフィールドは、図６２に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

【０２０３】

次に、図４５に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには２つのタイプがあり、それらはEP_mapとTU_mapである。図６３に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図６４に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。１つのAVストリームは、１つのEP_mapまたは一つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

【０２０４】

図６５は、CPIのシンタクスを示す図である。図６５に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す３２ビットの符号なし整数である。CPI_typeは、図６６に示すように、１ビットのフラグであり、ClipのCPIのタイプを表す。

【０２０５】

次に、図６５に示したCPIのシンタクス内のEP_mapについて説明する。EP_map

には、２つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが１つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【０２０６】

ビデオストリーム用のEP_mapについて図６７を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、および、RSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始めるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第１バイト目を含むソースポケットのアドレスを示す。

【０２０７】

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【０２０８】

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第１バイト目を含むソースポケットのアドレスを示す。

【０２０９】

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Cl

ipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【０２１０】

EP_mapとSTC_Infoの関係を説明するに、１つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく１つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図６８を参照）。

・EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、１つのEP_map_for_one_stream_PIDを持たねばならない。図６９に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

【０２１１】

図７０は、EP_mapのシンタクスを示す図である。図７０に示したEP_mapのシンタクスを説明するに、EP_typeは、４ビットのフィールドであり、図７１に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが１つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【０２１２】

number_of_stream_PIDsの１６ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム（ビデオまたはオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0 ('video')に等しい場合、そのエレメンタリストリームはビデオストリームでなけれならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

【０２１３】

num_EP_entries(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k)：この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第１バイト目からの大きさで示される。

【０２１４】

padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【０２１５】

図７２は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図７２に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。EP_typeが1 ('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。

【０２１６】

RSPN_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第１バイト目を含むソースポケットの相対アドレスを示す。または、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第一バイト目を含むソースポケットの相対アドレスを示す。

【０２１７】

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリ

ームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【０２１８】

次に、TU_mapについて、図７３を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、１つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【０２１９】

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

TU_start_time(k) = offset_time + k*time_unit_size

TU_start_time(k)は、45kHzの精度を持つ。

【０２２０】

図７５は、TU_mapのシンタクスを示す図である。図７５に示したTU_mapのシンタクスを説明するに、offset_timeの３２bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、２７MHz精度のアライバルタイムクロックから導き出される４５kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなければならない。

【０２２１】

time_unit_sizeの３２ビットフィールドは、time_unitの大きさを与えるものであり、それは２７MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、１秒以下（time_unit_size<=4500）にすることが良い。number_of_time_unit_entriesの３２ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリー数を示す。

【０２２２】

RSPN_time_unit_startの３２ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【０２２３】

図４５に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置１）によってセットされるものであり、ユーザによってセットされるものではない。

【０２２４】

図７５は、ClipMarkのシンタクスを示す図である。図７５に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【０２２５】

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark

()のバイト数を示す３２ビットの符号なし整数である。number_of_Clip_marksは、 ClipMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数。number_of_Clip_marks は、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図７６に示すテーブルに従って符号化される。

【０２２６】

mark_time_stampは、３２ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図７７に示すように、PlayList()の中のCPI_typeにより異なる。

【０２２７】

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この８ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この８ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

【０２２８】

name_lengthの８ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【０２２９】

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない。

【０２３０】

MakersPrivateDataについては、図２２を参照して既に説明したので、その説明は省略する。

【０２３１】

次に、サムネイルインフォメーション（Thumbnail Information）について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ１つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像，すなわちVolumeを代表する画像、および、それぞれのPlayListを代表する画像をストアする。すべてのメニューサムネイルは、ただ１つのmenu.thmbファイルにストアされる。

【０２３２】

mark.thmbファイルは、マークサムネイル画像，すなわちマーク点を表すピクチャをストアする。すべてのPlayListおよびClipに対するすべてのマークサムネイルは、ただ１つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は一つのtn_blockに格納される。１つの画像データはは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。１つのサムネイル画像のバイト長は可変である。

【０２３３】

図７８は、menu.thmbとmark.thmbのシンタクスを示す図であり、図７９は、図７８に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図７９に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す４個のキャラクター文字である。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。

【０２３４】

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersP

rivateData()のバイト数を示す３２ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す３２ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリー数を与える１６ビットの符号なし整数である。

【０２３５】

tn_block_sizeは、1024バイトを単位として、１つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size＝１ならば、それは１つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す１１６ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す１６ビットの符号なし整数である。thumbnail_index として、0xFFFFという値を使用してはならない。thumbnail_index はUIAppInfoVolume()、UIAppInfoPlayList()、 PlayListMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【０２３６】

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図８０に示すような値をとる。表中のDCFとPNGは”menu.thmb”内でのみ許される。マークサムネイルは、値”0x00”(MPEG-2 Video I-picture)をとらなければならない。

【０２３７】

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す３２ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す１６ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、０から始まり、tn_blockのfor-ループ中の変数ｋの値に関係する。

【０２３８】

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す１６ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す１６ビットの符号なし整数である。tn_blockは、 サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn_blockは、同じサイズ（固定長）であり、その大きさはtn_block_sizeによって定義される。

【０２３９】

図８１は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図８１のように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応する事ができるようになる。

【０２４０】

次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ（図１４）にストアされる。AVストリームファイルには、２つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【０２４１】

まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2 トランスポートストリームの構造は、図８２に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第１バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、４バイト長であり、またトランスポートパケットは、1

88バイト長である。

【０２４２】

１つのAligned unitは、３２個のソースパケットから成る。DVR MPEG２トランスポートストリームの中の最後のAligned unitも、また３２個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が３２の倍数でない時、ヌルパケット（PID=0x1FFFのトランスポートパケット）を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG２トランスポートストリームに余分な情報を付加してはならない。

【０２４３】

図８３に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図８３に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【０２４４】

MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリームまたはパーシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1（[2]）またはISO/IEC13818-9（[5]）に従っていなければならない。MPEG2トランスポートストリームのi番目のバイトは、T-STD(ISO/IEC13818-1で規定されるTransport stream system target decoder)とソースパケッタイザーへ、時刻t(i)に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

【０２４５】

２７MHz PLL５２は、２７MHzクロックの周波数を発生する。２７MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。arrival time clock counter５３は、２７MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【０２４６】

source packetizer５４は、すべてのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第１バイト目がT-STDとソースパケッタイザーの両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第１バイト目を示す。

$$arrival\_time\_stamp(k) = arrival\_time\_clock(k) \% 2^{30}$$

【０２４７】

２つの連続して入力されるトランスポートパケットの時間間隔が、$2^{30}/27000000$秒（約40秒）以上になる場合、その２つのトランスポートパケットのarrival_time_stampの差分は、$2^{30}/27000000$秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【０２４８】

smoothing buffer５５は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファは、オーバーフロウしてはならない。Rmaxは、スムージングバッファが空でない時のスムージングバッファからのソースパケットの出力ビットレートである。スムージングバッファが空である時、スムージングバッファからの出力ビットレートはゼロである。

【０２４９】

次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$Rmax = TS\_recording\_rate * 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【０２５０】

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recordi

ng_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター，例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【０２５１】

smoothing buffer sizeは、入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファの大きさはゼロである。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い。

【０２５２】

記録機（レコーダ）および再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。デフォールトのバッファサイズは、1536 bytes である。

【０２５３】

次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図８４は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【０２５４】

27MHz X-tal６１は、２７Mhzの周波数を発生する。２７MHz周波数の誤差範囲は、+/-30 ppm (27000000 +/- 810 Hz)でなければならない。arrival time clock counter６２は、２７MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【０２５５】

smoothing buffer６４において、Rmaxは、スムージングバッファがフルでない時のスムージングバッファへのソースパケットの入力ビットレートである。スム

ージングバッファがフルである時、スムージングバッファへの入力ビットレートはゼロである。

【０２５６】

MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB ３０ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スムージングバッファから引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファは、アンダーフロウしてはならない。

【０２５７】

DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

【０２５８】

図８５は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図８５に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図８６に示す。図８６に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、またはcopy prohibitedとすることができる。図８７は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

【０２５９】

copy_permission_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセル

フエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【０２６０】

arrival_time_stampは、次式

$$arrival\_time\_stamp(k) = arrival\_time\_clock(k) \% 2^{30}$$

において、arrival_time_stampによって指定される値を持つ整数値である。

【０２６１】

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース（STCベース）の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

【０２６２】

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、２６時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース（STCベース）の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間).と規定している。

【０２６３】

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、１つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならず、かつ後述するDVR-STDに従わなければならない。

【０２６４】

本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシ

ームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフロウを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

【０２６５】

"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【０２６６】

シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【０２６７】

図８８は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図８８においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図８８に示したTS1は、Clip1（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

【０２６８】

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図８８においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられ

たストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【０２６９】

また、図８８におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time（図８８においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【０２７０】

図８９は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図８９におけるTS1は、Clip1（Clip AVストリーム)の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図８９においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図８９におけるTS2は、Clip2（Clip AVストリーム)の影を付けられたストリームデータから成る。

【０２７１】

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time（図８９においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【０２７２】

図８８と図８９において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、１でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、１でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、２以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および／またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【０２７３】

ビデオビットストリームの制限について説明する。図９０は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1（Clip1のOUT_time）の後とIN_time2（Clip2のIN_time）の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【０２７４】

図９０に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図９１に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図９０のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、１つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【０２７５】

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図９０のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始す

る事ができて、これに続くClip2のビデオストリームに接続され、１つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【０２７６】

図９０に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図９２に示す。Clip1のビデオストリームは、図９０のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、１つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図９０のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG２規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【０２７７】

ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【０２７８】

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップーボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、”top_field_first” および ”repeat_first_field”フラグを書き換える必要があるかもしれない，またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【０２７９】

オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例．MPEG1レイヤ2，AC-3，SESF LPCM，AAC）は、同じでなければならない。

【０２８０】

次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【０２８１】

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図９３に示すように、２オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあっても良い。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【０２８２】

ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【０２８３】

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ１つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【０２８４】

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【０２８５】

PlayItemの制限について説明するに、先行するPlayItemのOUT_time（図８８、図８９において示されるOUT_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time（F図８８、図８９において示されるIN_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【０２８６】

Bridge-Clipを使用する場合のデータアロケーションの制限について、図９４を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）とClip2（Clip AVストリームファイル）に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【０２８７】

RSPN_exit_from_previous_Clip以前のClip1（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【０２８８】

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図９５を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）の最後の部分とClip2（Clip AVストリームファイル）の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【０２８９】

Clip1（Clip AVストリームファイル）の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2（Clip AVストリームファイル）の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【０２９０】

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランスポートストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルでもある。

【０２９１】

DVR-STDモデルを図９６に示す。図９６に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n，TBn，MBn，EBn，TBsys，Bsys，Rxn，Rbxn，Rxsys，Dn，Dsys，OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファでる。

【０２９２】

MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号

中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【０２９３】

Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【０２９４】

DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1，TBnまたはTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1，MB1，EB1，TBn，Bn，TBsysおよびTBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【０２９５】

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される２つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図８８に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【０２９６】

図９７は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力，復号，表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイム

ベースの時間軸（図９７においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図９７においてATC1で示される）と同じでない。

【０２９７】

また、TS2のシステムタイムベースの時間軸（図９７においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図９７においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【０２９８】

DVR-STD への入力タイミングについて説明する。時刻T_1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn またはTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【０２９９】

TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBnまたはTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T_2である。従って、時刻T_1からT_2までの区間では、ソースパケットのarrival_time_stampは無視される。

【０３００】

N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト数とすると、時刻T_1乃至T_2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$DT1 = T_2 - T_1 = N1 / TS\_recording\_rate\ (TS1)$

時刻T_1乃至T_2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

【０３０１】

T_2の時刻において、arrival time clock counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1，TBn またはTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

【０３０２】

付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻Ｔ１からT2までの区間の入力データを処理することができるように、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【０３０３】

ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図９７ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図９７ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

【０３０４】

STC1とSTC2の間のオフセットは、次のように決定される。PTS^1_{end}は、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS^2_{start}は、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、T_{pp}は、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、２つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC\_delta = PTS^1_{end} + T_{pp} - PTS^2_{start}$$

【０３０５】

オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは０乃至２オーディオフレーム未満である（図９７に

図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

【０３０６】

DVR-STDのシステムタイムクロックについて説明するに、時刻T_5において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻T_2からT_5の間にオーバーラップしていても良い。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値（STC1）と新しいタイムベースの値（STC2）の間で切り替える。STC2の値は、次式により算出される。

$$STC2 = STC1 - STC\_delta$$

【０３０７】

バッファリングの連続性について説明する。${STC1}^1_{video_end}$は、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC1上のSTCの値である。${STC2}^2_{video_start}$は、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC2上のSTCの値である。${STC2}^1_{video_end}$は、${STC1}^1_{video_end}$ の値をシステムタイムベースSTC2上の値に換算した値である。${STC2}^1_{video_end}$は、次式により算出される。

$${STC2}^1_{video\_end} = {STC1}^1_{video\_end} - STC\_delta$$

【０３０８】

DVR-STDに従うために、次の２つの条件を満たす事が要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$${STC2}^2_{video\_start} > {STC2}^1_{video\_end} + \Delta T_1$$

この不等式が満たされるように、Clip１および、または、Clip２の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【０３０９】

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフロウおよびアンダーフローさせてはならない。

【０３１０】

このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【０３１１】

上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【０３１２】

この記録媒体は、図９８に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク２２１（フロッピディスクを含む）、光ディスク２２２（CD-ROM（Compact Disk-Read Only Memory），DVD（Digital Versatile Disk）を含む）、光磁気ディスク２２３（MD（Mini-Disk）を含む）、若しくは半導体メモリ２２４などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM２０２や記憶部２０８が含まれるハードディスクなどで構成される。

【０３１３】

なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むもので

ある。

【０３１４】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【０３１５】

【発明の効果】

以上の如く、請求項１に記載の情報処理装置、請求項３に記載の情報処理方法、および、請求項４に記載の記録媒体によれば、AVストリームの記録が開始された時点から終了される時点までに記録されたAVストリームを１単位とし、単位毎に、AVストリームの属性情報として、AVストリームの記録モードに関する情報、AVストリームの記録レートの平均値に関する情報、AVストリームのうち符号化情報が同一である区間に関する情報、AVストリーム中のランダムアクセス可能な位置に関する情報、AVストリーム中の特徴的な画像に関する情報のうち、少なくとも１つの情報を記録媒体に記録するようにしたので、その情報を用いることにより、AVストリームの読み出し位置の決定や復号処理が速やかに行うことが可能となる。

【図面の簡単な説明】

【図１】

本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図２】

記録再生装置１により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図３】

Real PlayListとVirtual PlayListについて説明する図である。

【図４】

Real PlayListの作成について説明する図である。

【図５】

Real PlayListの削除について説明する図である。

【図６】

アセンブル編集について説明する図である。

【図７】

Virtual PlayListにサブパスを設ける場合について説明する図である。

【図８】

PlayListの再生順序の変更について説明する図である。

【図９】

PlayList上のマークとClip上のマークについて説明する図である。

【図１０】

メニューサムネイルについて説明する図である。

【図１１】

PlayListに付加されるマークについて説明する図である。

【図１２】

クリップに付加されるマークについて説明する図である。

【図１３】

PlayList、Clip、サムネイルファイルの関係について説明する図である。

【図１４】

ディレクトリ構造について説明する図である。

【図１５】

info.dvrのシンタクスを示す図である。

【図１６】

DVR volumeのシンタクスを示す図である。

【図１７】

Resumevolumeのシンタクスを示す図である。

【図１８】

UIAppInfovolumeのシンタクスを示す図である。

【図１９】

Character set valueのテーブルを示す図である。

【図２０】

TableOfPlayListのシンタクスを示す図である。

【図２１】

TableOfPlayListの他のシンタクスを示す図である。

【図２２】

MakersPrivateDataのシンタクスを示す図である。

【図２３】

xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

【図２４】

PlayListについて説明する図である。

【図２５】

PlayListのシンタクスを示す図である。

【図２６】

PlayList_typeのテーブルを示す図である。

【図２７】

UIAppinfoPlayListのシンタクスを示す図である。

【図２８】

図２７に示したUIAppinfoPlayListのシンタクス内のフラグについて説明する図である。

【図２９】

PlayItemについて説明する図である。

【図３０】

PlayItemについて説明する図である。

【図３１】

PlayItemについて説明する図である。

【図３２】

PlayItemのシンタクスを示す図である。

【図３３】

IN_timeについて説明する図である。

【図３４】

OUT_timeについて説明する図である。

【図３５】

Connection_Conditionのテーブルを示す図である。

【図３６】

Connection_Conditionについて説明する図である。

【図３７】

BridgeSequenceInfoを説明する図である。

【図３８】

BridgeSequenceInfoのシンタクスを示す図である。

【図３９】

SubPlayItemについて説明する図である。

【図４０】

SubPlayItemのシンタクスを示す図である。

【図４１】

SubPath_typeのテーブルを示す図である。

【図４２】

PlayListMarkのシンタクスを示す図である。

【図４３】

Mark_typeのテーブルを示す図である。

【図４４】

Mark_time_stampを説明する図である。

【図４５】

zzzzz.clipのシンタクスを示す図である。

【図４６】

ClipInfoのシンタクスを示す図である。

【図４７】

Clip_stream_typeのテーブルを示す図である。

【図４８】

offset_SPNについて説明する図である。

【図４９】

offset_SPNについて説明する図である。

【図５０】

ＳＴＣ区間について説明する図である。

【図５１】

STC_Infoについて説明する図である。

【図５２】

STC_Infoのシンタクスを示す図である。

【図５３】

ProgramInfoを説明する図である。

【図５４】

ProgramInfoのシンタクスを示す図である。

【図５５】

VideoCondingInfoのシンタクスを示す図である。

【図５６】

Video_formatのテーブルを示す図である。

【図５７】

frame_rateのテーブルを示す図である。

【図５８】

display_aspect_ratioのテーブルを示す図である。

【図５９】

AudioCondingInfoのシンタクスを示す図である。

【図６０】

audio_codingのテーブルを示す図である。

【図６１】

audio_component_typeのテーブルを示す図である。

【図６２】

sampling_frequencyのテーブルを示す図である。

【図６３】

CPIについて説明する図である。

【図６４】

CPIについて説明する図である。

【図６５】

CPIのシンタクスを示す図である。

【図６６】

CPI_typeのテーブルを示す図である。

【図６７】

ビデオEP_mapについて説明する図である。

【図６８】

EP_mapについて説明する図である。

【図６９】

EP_mapについて説明する図である。

【図７０】

EP_mapのシンタクスを示す図である。

【図７１】

EP_type valuesのテーブルを示す図である。

【図７２】

EP_map_for_one_stream_PIDのシンタクスを示す図である。

【図７３】

TU_mapについて説明する図である。

【図７４】

TU_mapのシンタクスを示す図である。

【図７５】

ClipMarkのシンタクスを示す図である。

【図７６】

mark_typeのテーブルを示す図である。

【図７７】

mark_type_stampのテーブルを示す図である。

【図７８】

menu.thmbとmark.thmbのシンタクスを示す図である。

【図７９】

Thumbnailのシンタクスを示す図である。

【図８０】

thumbnail_picture_formatのテーブルを示す図である。

【図８１】

tn_blockについて説明する図である。

【図８２】

DVR MPEG２のトランスポートストリームの構造について説明する図である。

【図８３】

DVR MPEG２のトランスポートストリームのレコーダモデルを示す図である。

【図８４】

DVR MPEG２のトランスポートストリームのプレーヤモデルを示す図である。

【図８５】

source packetのシンタクスを示す図である。

【図８６】

TP_extra_headerのシンタクスを示す図である。

【図８７】

copy permission indicatorのテーブルを示す図である。

【図８８】

シームレス接続について説明する図である。

【図８９】

シームレス接続について説明する図である。

【図９０】

シームレス接続について説明する図である

【図９１】

シームレス接続について説明する図である。

【図９２】

シームレス接続について説明する図である

【図９３】

オーディオのオーバーラップについて説明する図である。

【図９４】

BridgeSequenceを用いたシームレス接続について説明する図である。

【図９５】

BridgeSequenceを用いないシームレス接続について説明する図である。

【図９６】

DVR STDモデルを示す図である。

【図９７】

復号、表示のタイミングチャートを示す図である。

【図９８】

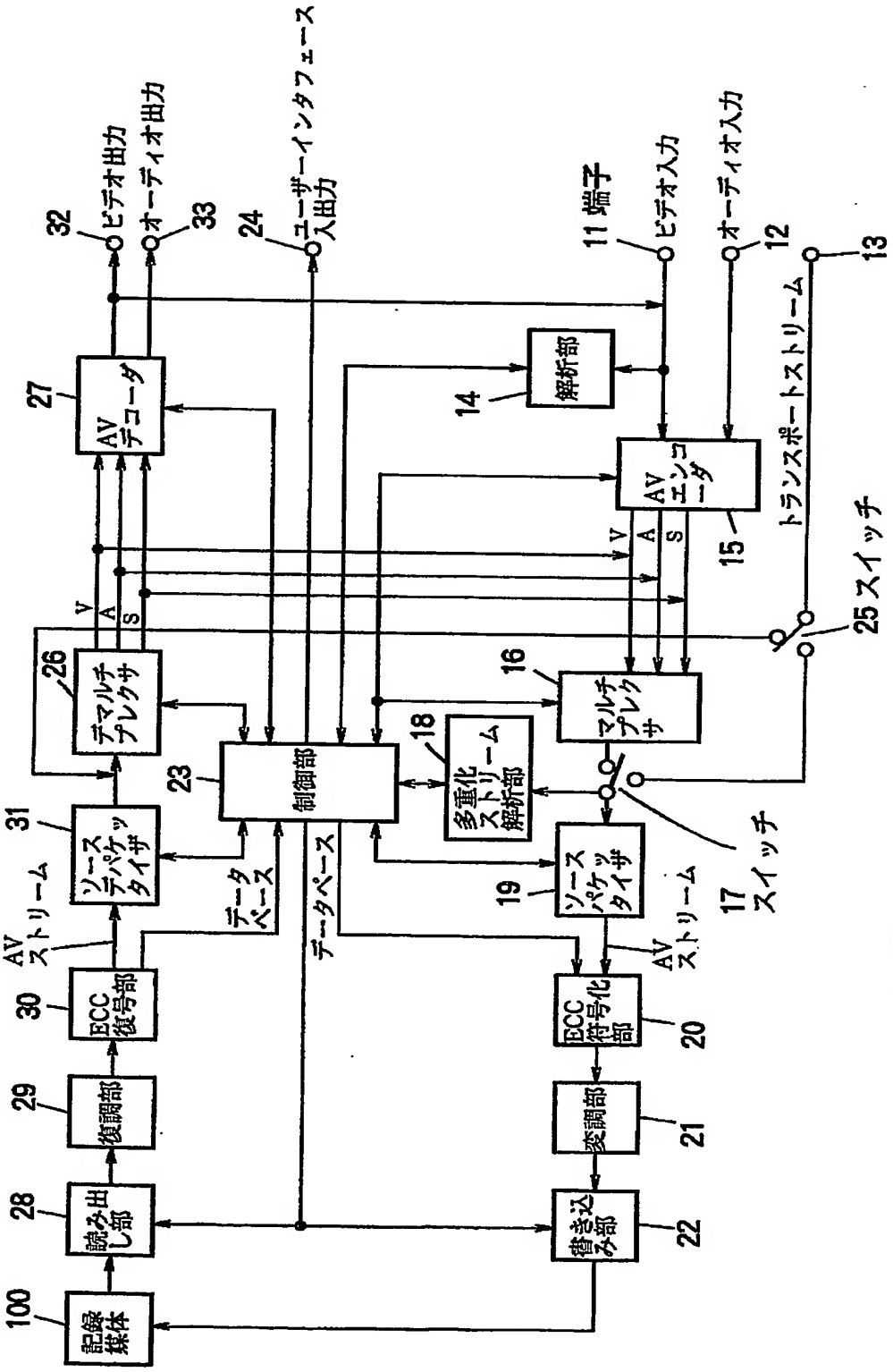
媒体を説明する図である。

【符号の説明】

１　記録再生装置，　１１乃至１３　端子，　１４　解析部，　１５　AVエンコーダ，　１６　マルチプレクサ，　１７　スイッチ，　１８　多重化ストリーム解析部，　１９　ソースパケッタイザ，　２０　ECC符号化部，　２１　変調部，　２２　書き込み部，　２３　制御部，　２４　ユーザインタフェース，　２５　スイッチ，　２６　デマルチプレクサ，　２７　AVデコーダ，　２８　読み出し部，　２９　復調部，　３０　ECC復号部，　３１　ソースパケッタイザ，　３２，３３　端子

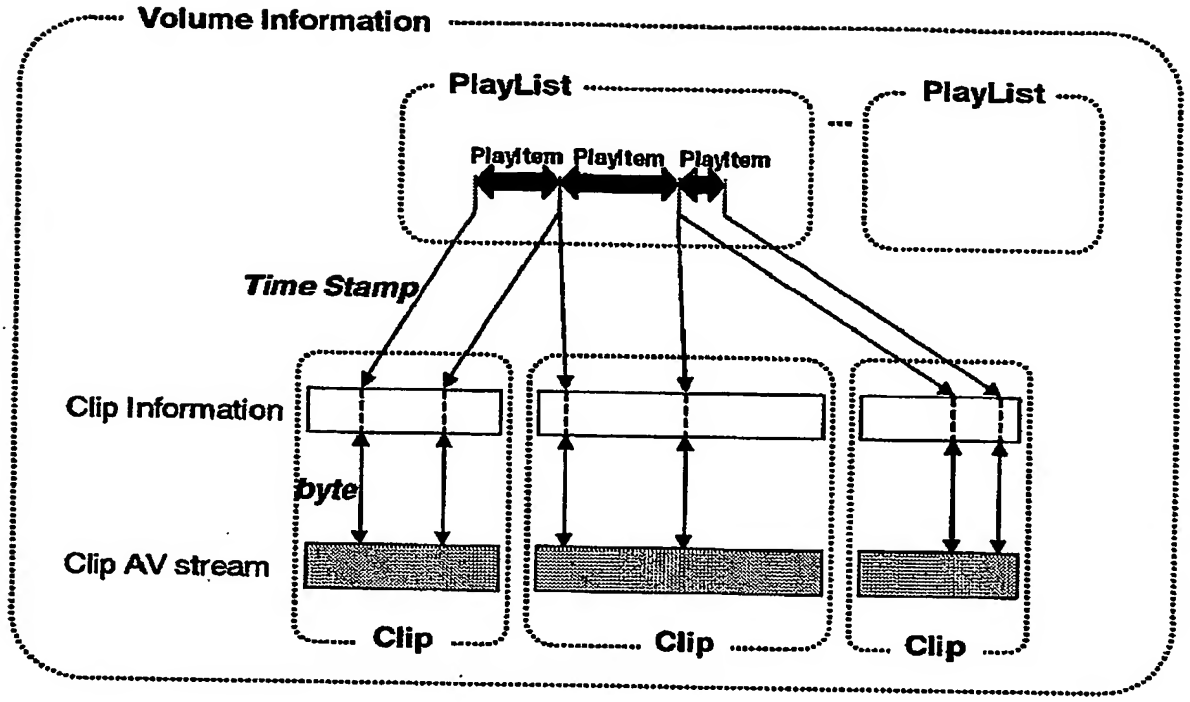
【書類名】　図面

【図１】

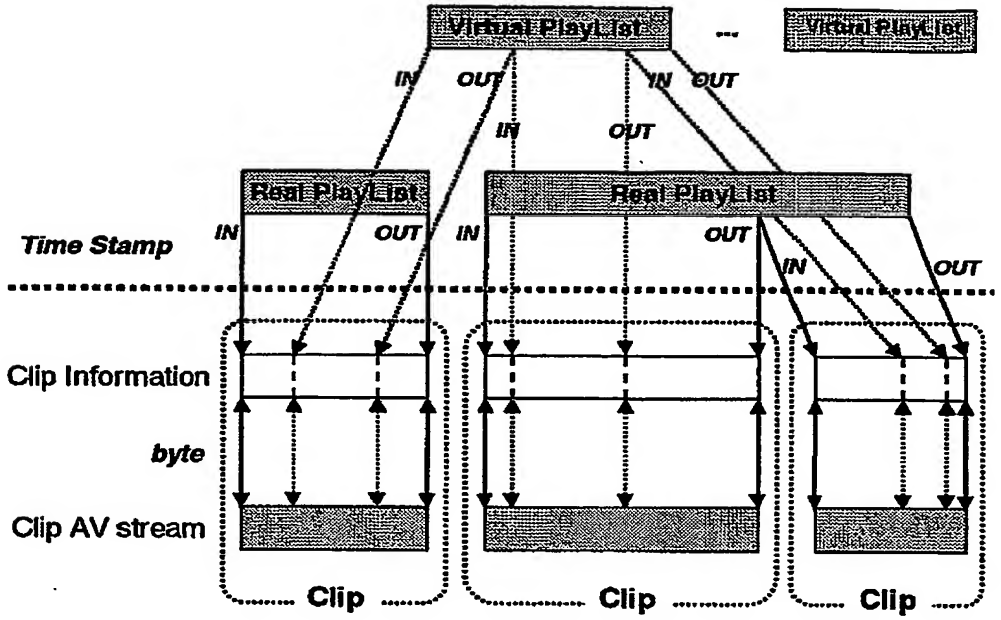


記録再生装置１

【図２】

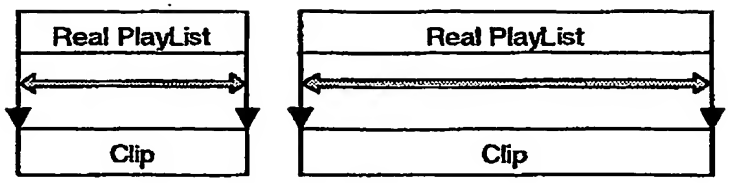

Volume Information
PlayList
PlayList
PlayItem
PlayItem
PlayItem
Time Stamp
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

【図３】


Virtual PlayList
Virtual PlayList
IN
OUT
IN
OUT
IN
OUT
Real PlayList
Real PlayList
Time Stamp
IN
OUT
IN
OUT
IN
OUT
Clip Information
byte
Clip AV stream
Clip
Clip
Clip

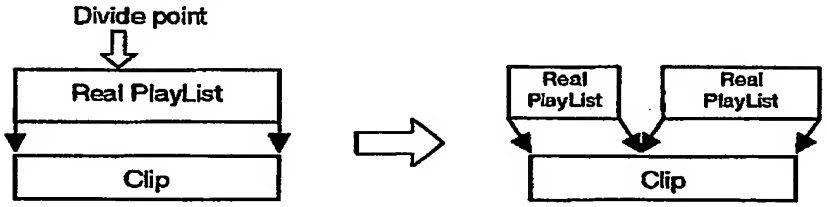
【図４】

(A)



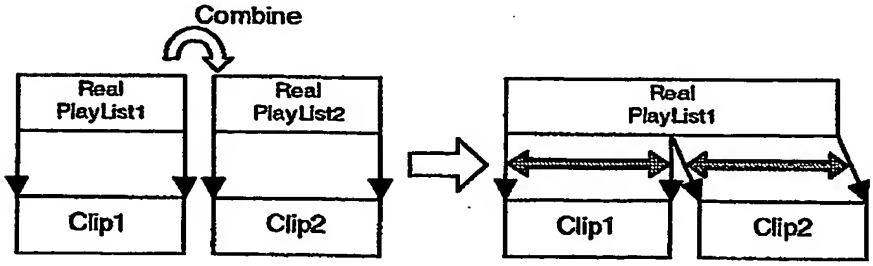
Real PlayListのクリエイトの例

(B)



Real PlayListのディバイドの例

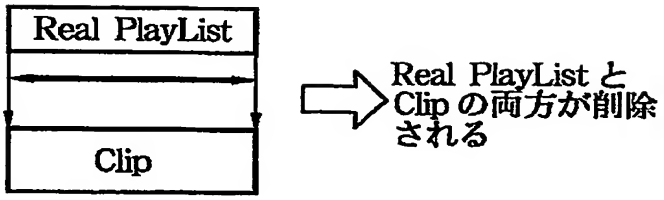
(C)



Real PlayListのコンバインの例

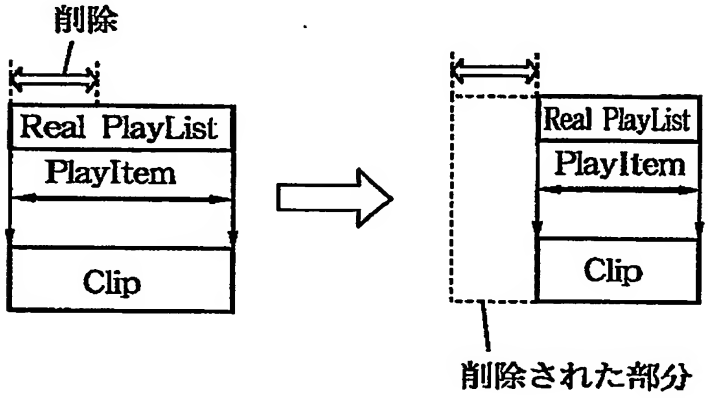
【図５】

(A)



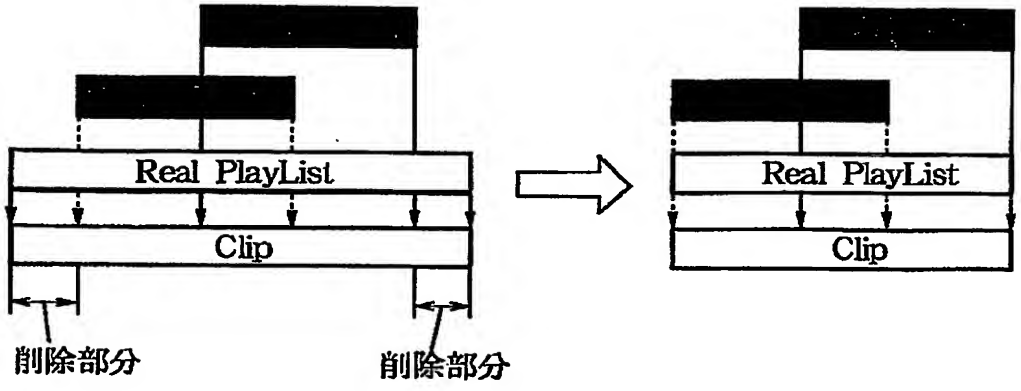
Real PlayList 全体のデリートの例

(B)



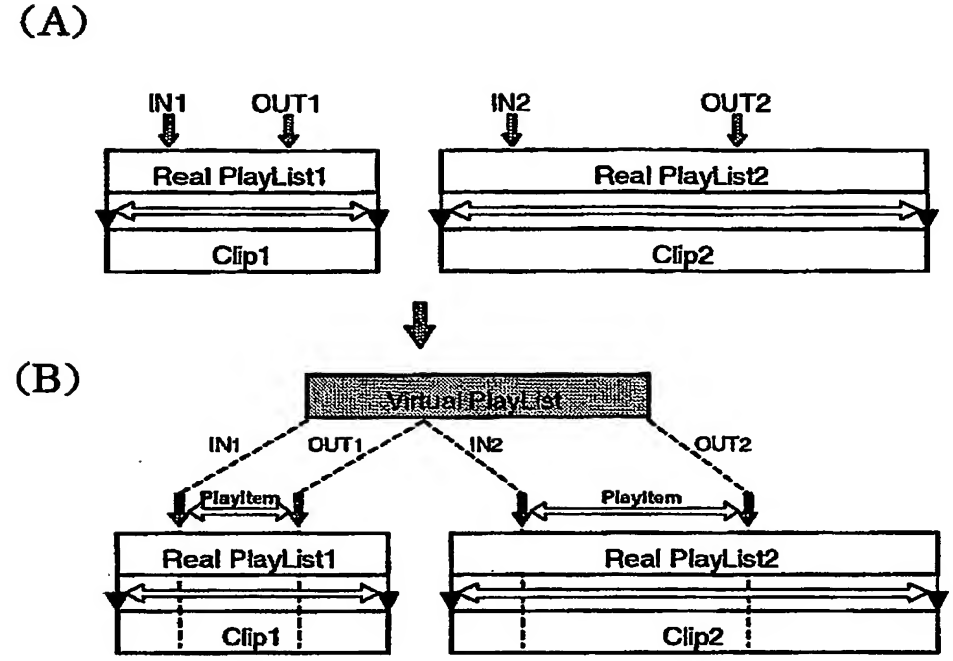
Real PlayList の部分的なデリートの例

(C)



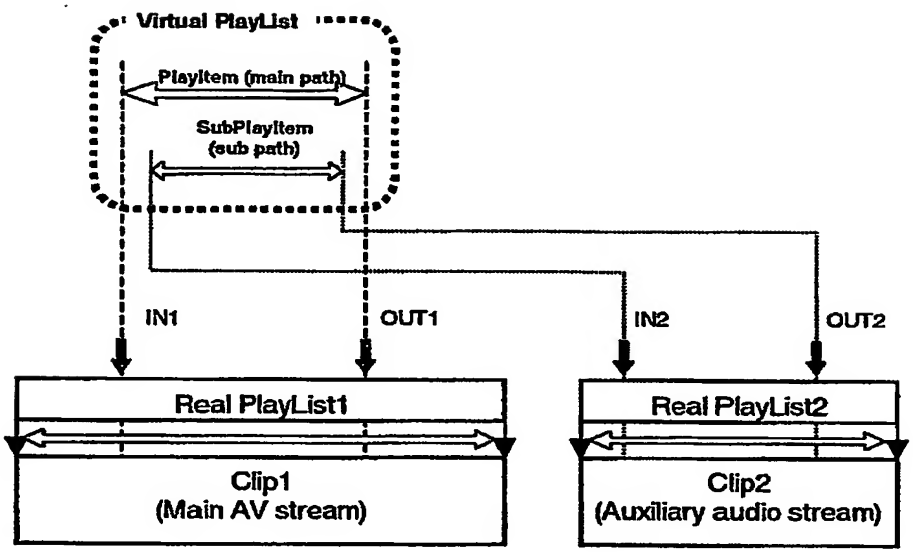
Real PlayList のミニマイズの例

【図６】



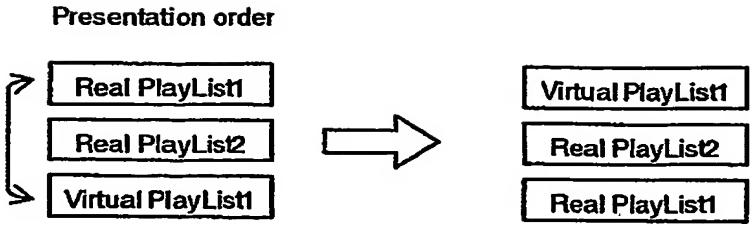
アセンブル編集の例

【図７】



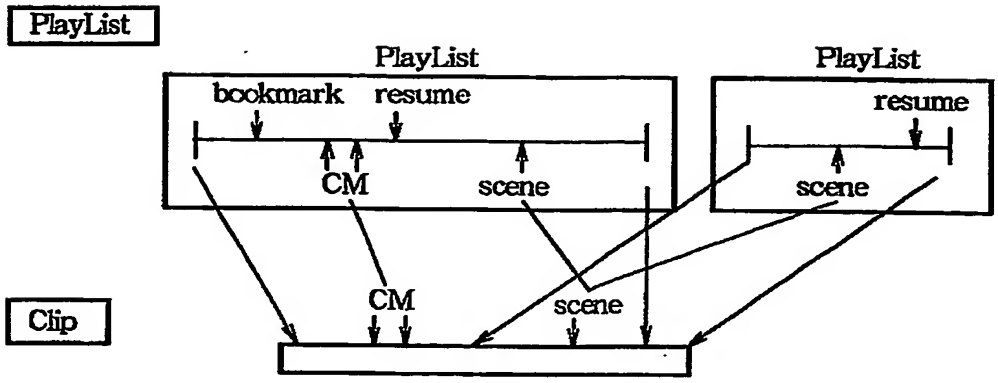
Virtual PlayList へのオーディオのアフレコの例

【図８】



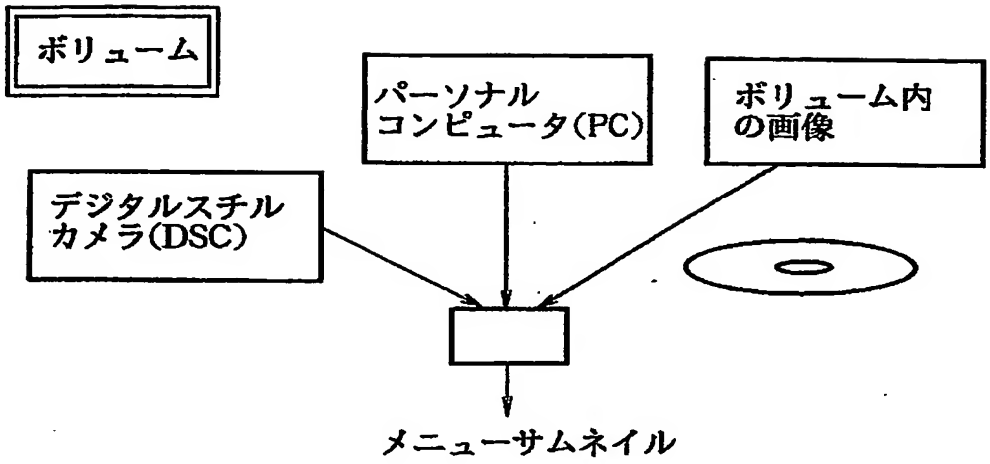
PlayList の再生順序の変更の例

【図９】

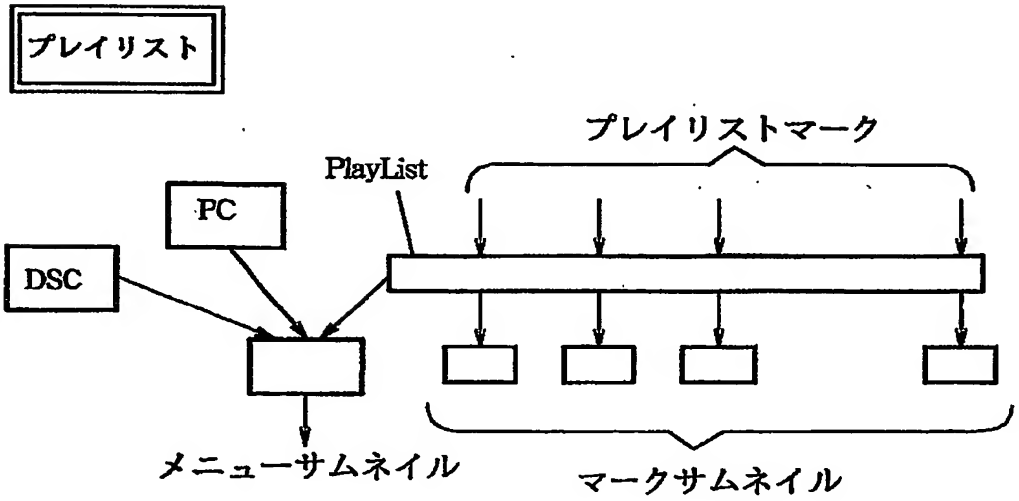


Playlist 上のマークと Clip 上のマーク

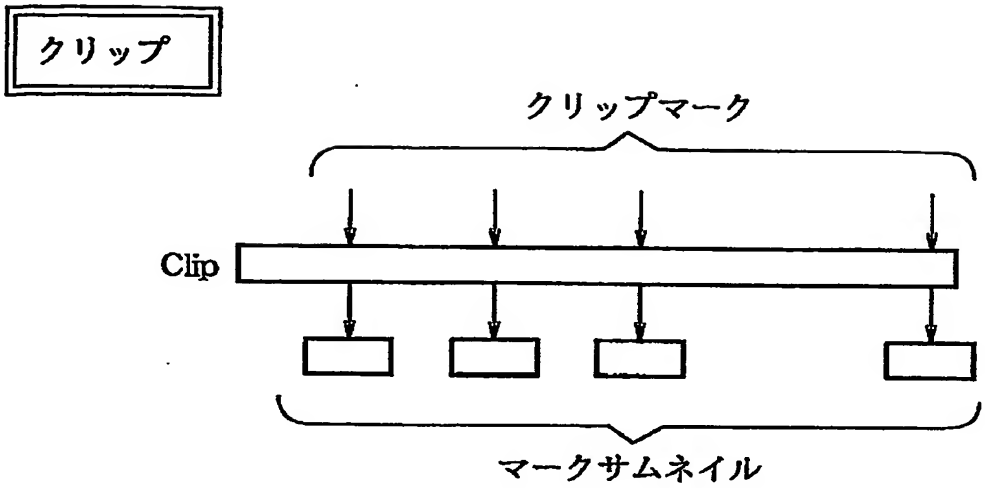
【図１０】


ボリューム
パーソナル
コンピュータ(PC)
ボリューム内
の画像
デジタルスチル
カメラ(DSC)
メニューサムネイル

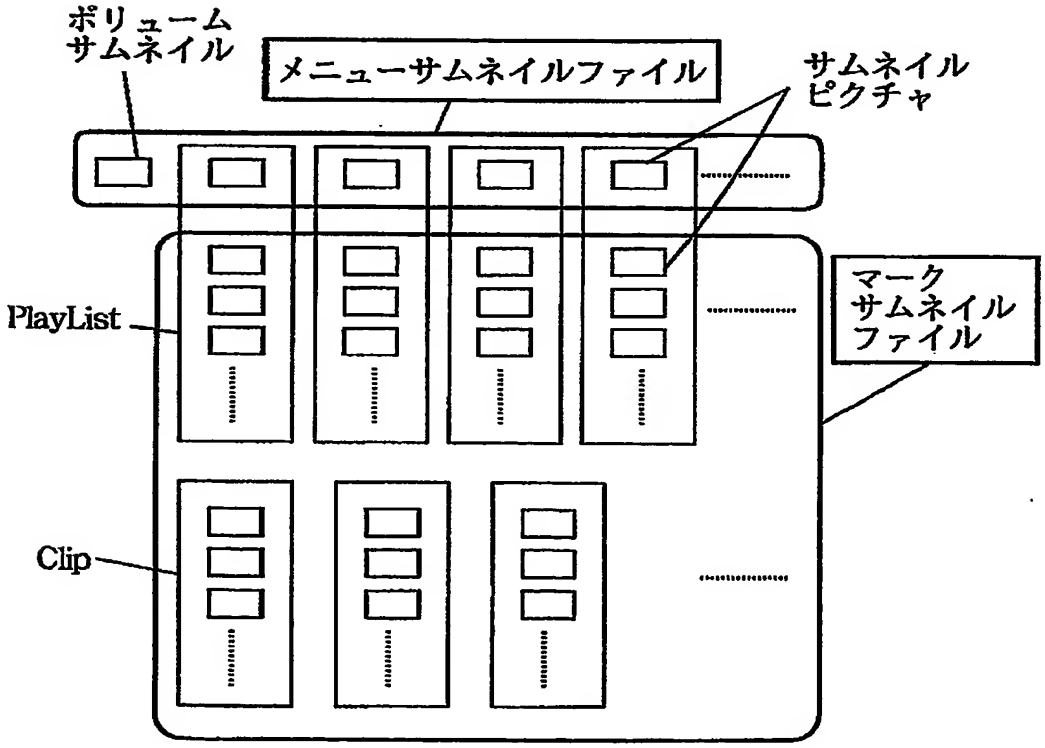
【図１１】


プレイリスト
プレイリストマーク
PlayList
PC
DSC
メニューサムネイル
マークサムネイル

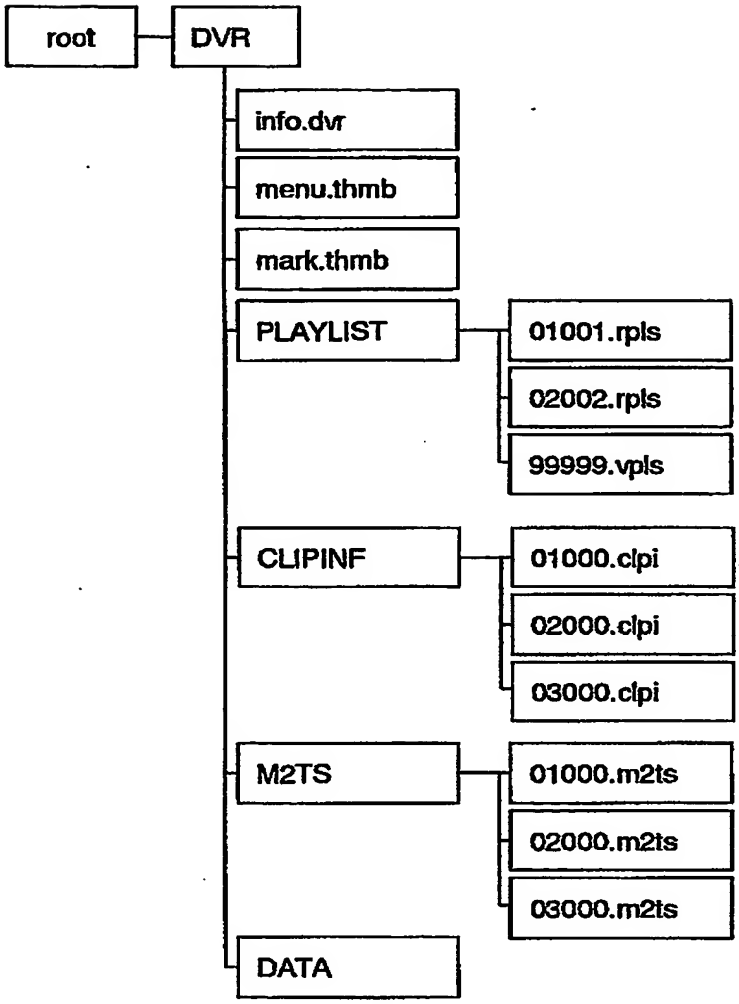
【図１２】


クリップ
クリップマーク
Clip
マークサムネイル

【図１３】


ボリューム
サムネイル
メニューサムネイルファイル
サムネイル
ピクチャ
PlayList
マーク
サムネイル
ファイル
Clip

【図１４】


root
DVR
info.dvr
menu.thmb
mark.thmb
PLAYLIST
01001.rpls
02002.rpls
99999.vpls
CLIPINF
01000.clpi
02000.clpi
03000.clpi
M2TS
01000.m2ts
02000.m2ts
03000.m2ts
DATA

【図１５】

Syntax	No. of bits	Mnemonics
info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

info.drv のシンタクス

【図１６】

Syntax	No. of bits	Mnemonics
DVRVolume() {		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UIAppInfoVolume()		
}		

DVR Volume のシンタクス

【図１７】

Syntax	No. of bits	Mnemonics
ResumeVolume() {		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

ResumeVolumeのシンタクス

【図１８】

Syntax	No. of bits	Mnemonics
UIAppInfoVolume () {		
character_set	8	bslbf
name_length	8	uimsbf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoVolumeのシンタクス

【図１９】

Value	Character coding
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

Character set value

【図２０】

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number_of_PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図２１】

■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayLists	16	uimsbf
for (i=0; i<*number of PlayLists*; i++) {		
PlayList_file_name	8*10	bslbf
UIAppInfoPlayList()		
}		
}		

TableOfPlayLists の別シンタクス

Syntax	No. of bits	Mnemonics
MakersPrivateData() {		
version_number	8*4	bslbf
length	32	uimsbf
if(length !=0){		
mpd_blocks_start_address	32	uimsbf
number_of_maker_entries	16	uimsbf
mpd_block_size	16	uimsbf
number_of_mpd_blocks	16	uimsbf
reserved	16	bslbf
for (i=0; i<*number_of_maker_entries*; i++){		
maker_ID	16	uimsbf
maker_model_code	16	uimsbf
start_mpd_block_number	16	uimsbf
reserved	16	bslbf
mpd_length	32	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for (j=0; j<*number_of_mpd_blocks*; j++) {		
mpd_block	mpd_block_size*1024*8	
}		
}		
}		

MakersPrivateData のシンタクス

特２０００－１８３７７０

出証特２００１－３０１４４２９

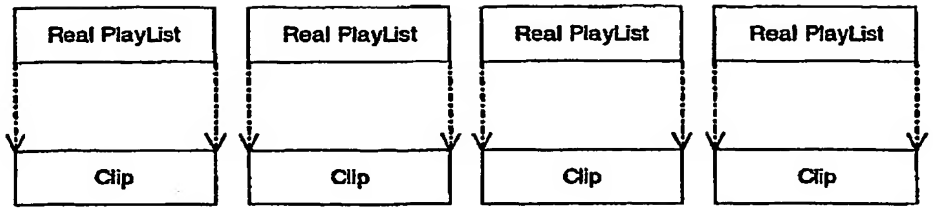
【図２３】

Syntax	No. of bits	Mnemonics
xxxxx.rpls / yyyyy.vpls {		
PlayListMark_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
PlayList()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

xxxxx.rpls と yyyyy.vpls のシンタクス

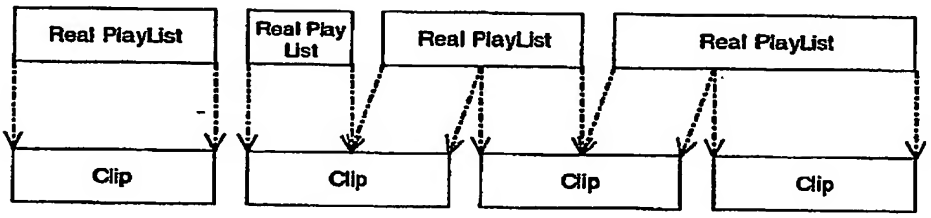
【図２４】

(A)



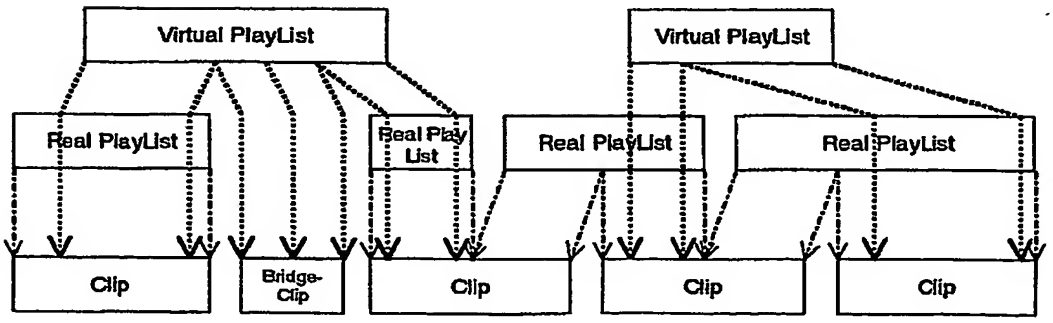
初めてAVストリームがClipとして記録された時のReal PlayListの例

(B)



編集後のReal PlayListの例

(C)



Virtual PlayListの例

Syntax	No. of bits	Mnemonics
PlayList() {		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList()		
number_of_PlayItems // main path	16	uimsbf
if (*<Virtual PlayList>*) {		
number_of_SubPlayItems // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (*PlayItem_id=0;* *PlayItem_id<number_of_PlayItems;* *PlayItem_id++*) {		
PlayItem() // main path		
}		
if (*<Virtual PlayList>*) {		
if (CPI_type==0 && PlayList_type==0) {		
for (i = 0; i < *number_of_SubPlayItems*; i++)		
SubPlayItem() // sub path		
}		
}		
}		

PlayListのシンタクス

出証特２００１－３０１４４２９

特２０００－１８３７７０

【図２６】

PlayList_type	Meaning
0	**AV記録のための PlayList** この PlayList に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	**オーディオ記録のための PlayList** この PlayList に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2 - 255	reserved

PlayList_type

Syntax	No. of bits	Mnemonics
UIAppInfoPlayList2(){		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoPlayList のシンタクス

特２０００－１８３７７０

出証特２００１－３０１４４２

【図２８】

(A)

write_protect_flag	Meaning
0b	その PlayList を自由に消去しても良い。
1b	write_protect_flag を除いてその PlayList の内容は、消去および変更されるべきではない。

write_protect_flag

(B)

is_played_flag	Meaning
0b	その PlayList は、記録されてから一度も再生されたことがない。
1b	PlayList は、記録されてから一度は再生された。

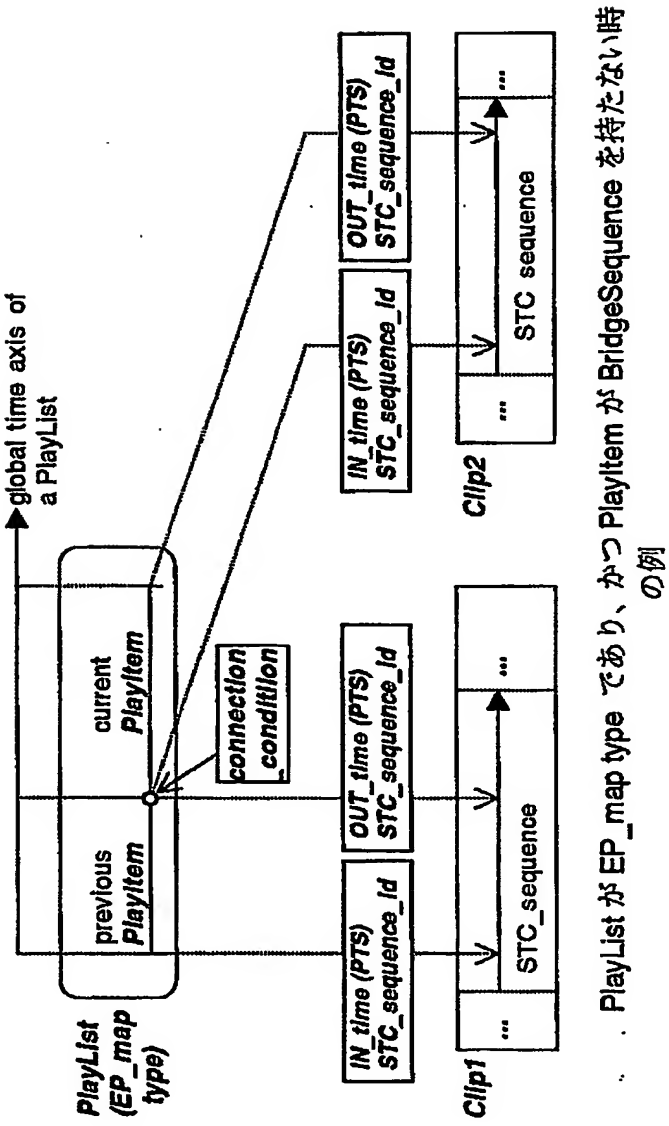
is_played_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

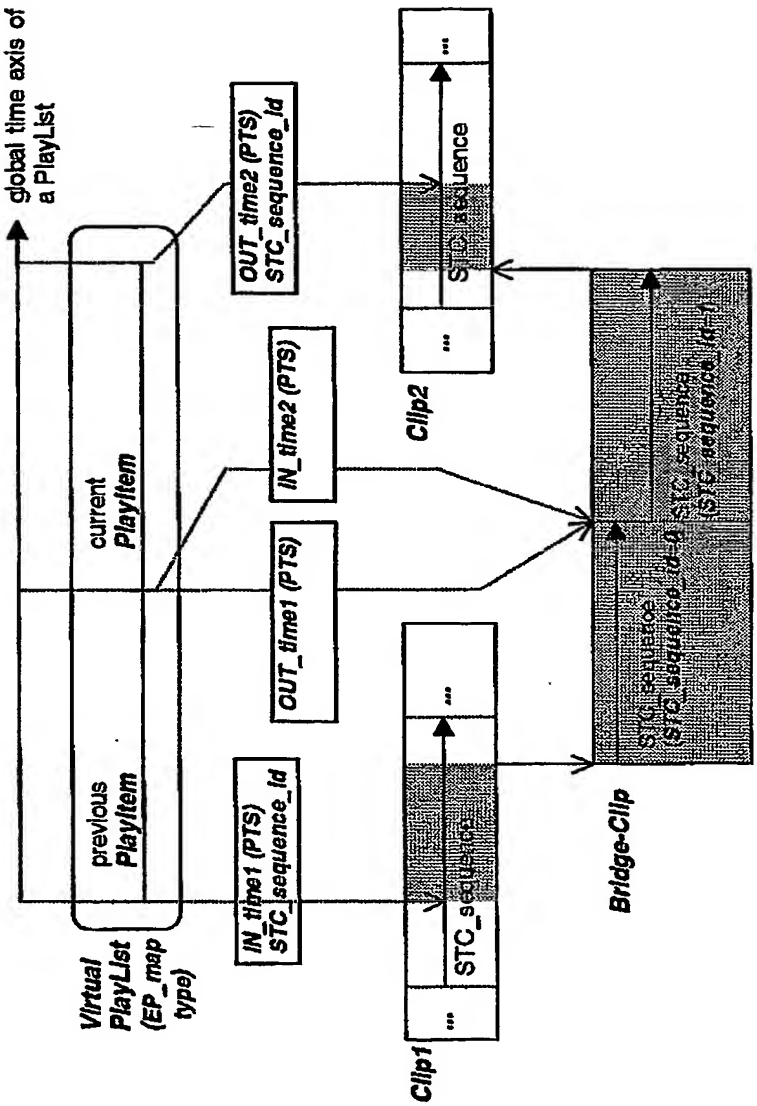
archive

【図２９】



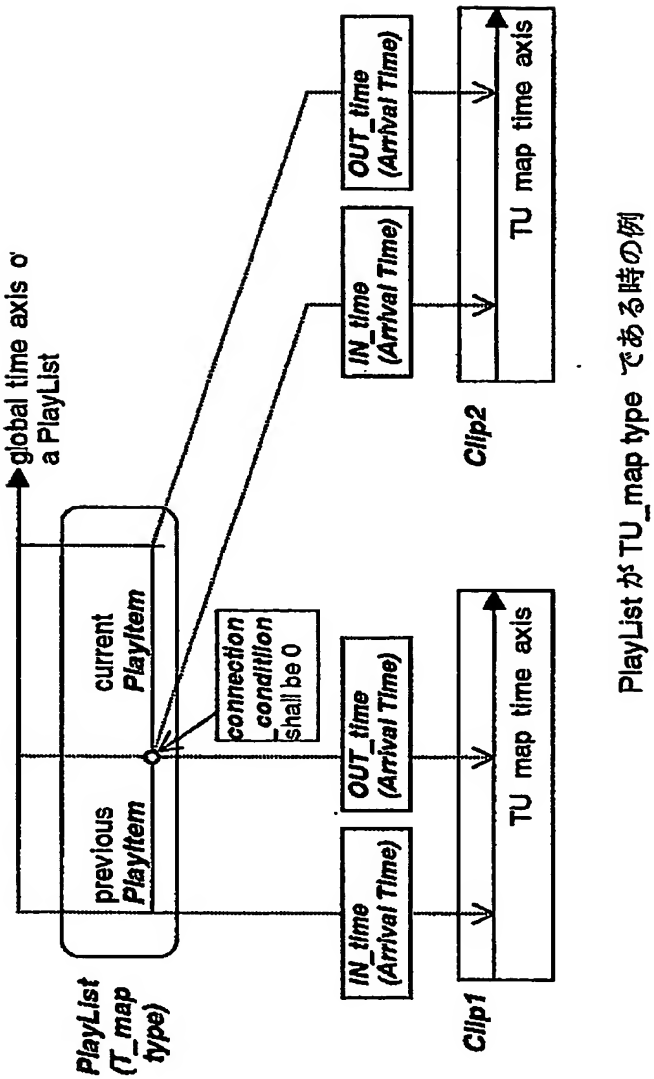
PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図３０】



・PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持つ時の例

【図３１】



PlayList が TU_map type である時の例

【図３２】

Syntax	No. of bits	Mnemonics
PlayItem() {		
Clip_information_file_name	8*10	bslbf
reserved	24	bslbf
STC_sequence_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
reserved	14	bslbf
connection_condition	2	bslbf
if (<Virtual PlayList>) {		
if (*connection_condition=='10'*) {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItemのシンタクス

【図３３】

CPI_type in the PlayList()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	IN_time は、*TU_map_time_axis* 上の時刻でなければならない。かつ、IN_time は、time_unit の精度に丸めて表さねばならない。IN_time は、次に示す等式により計算される。 $IN\_time = TU\_start\_time \% 2^{32}$

IN_time

CPI_type in the PlayList()	Semantics of OUT_time
EP_map type	OUT_time は、次に示す等式によって計算される Presenation_end_TS の値の上位32ビットを示さなければならない。 ***Presentation_end_TS = PTS_out + AU_duration*** ここで、 *PTS_out* は、PlayItem の中で最後のプレゼンテーションユニットに対応する33ビット長の PTS である。 *AU_duration* は、最後のプレゼンテーションユニットの 90kHz 単位の表示期間である。
TU_map type	OUT_time は、*TU_map_time_axis* 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さねばならない。OUT_time は、次に示す等式により計算される。 $OUT\_time = TU\_start\_time \% 2^{32}$

OUT_time

特２０００－１８３７７０

出証特２００１－３０１４４２９

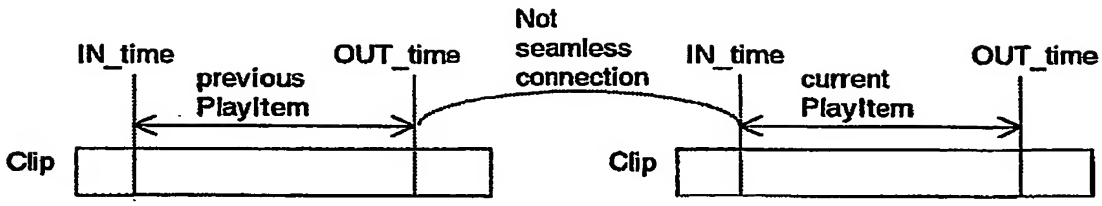
【図３５】

connection_condition	meaning
00	• 先行するPlayItemと現在のPlayItemの接続は、シームレス再生の保証がなされていない。 • PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされねばならない。
01	• この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 • 先行するPlayItemと現在のPlayItemは、システムタイムベース（STC ベース）の不連続点があるために分割されていることを表す。
10	• この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 • この状態は、Virtual PlayListに対してだけ許される。 • 先行するPlayItemと現在のPlayItemとの接続は、シームレス再生の保証がなされている。 • 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述するDVR-STDに従っていなければならない。
11	• この状態は、PlayListのCPI_typeがEP_map typeである場合にだけ許される。 • 先行するPlayItemと現在のPlayItemは、シームレス再生の保証がなされている。 • 先行するPlayItemと現在のPlayItemは、BridgeSequenceを使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述するDVR-STDに従っていなければならない。

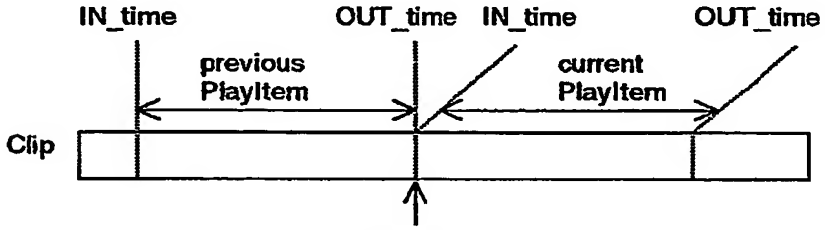
connection__condition

【図３６】

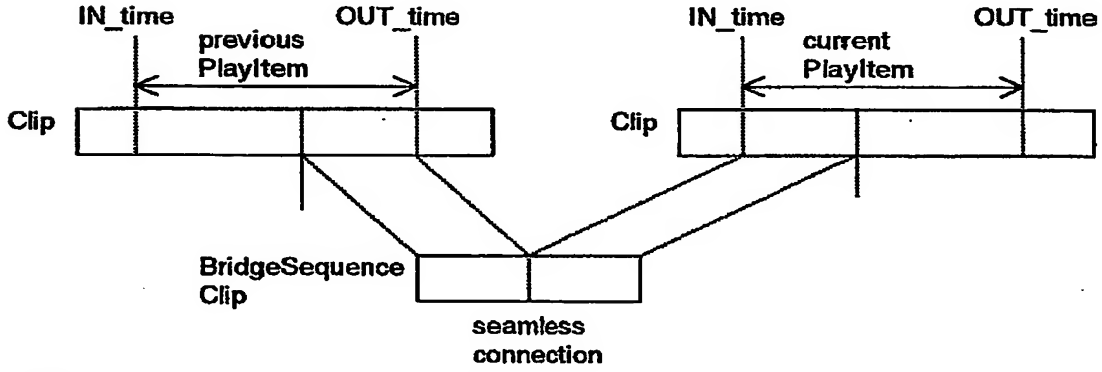
(A)



(B) connection_condition='00'



(C) connection_condition='01'

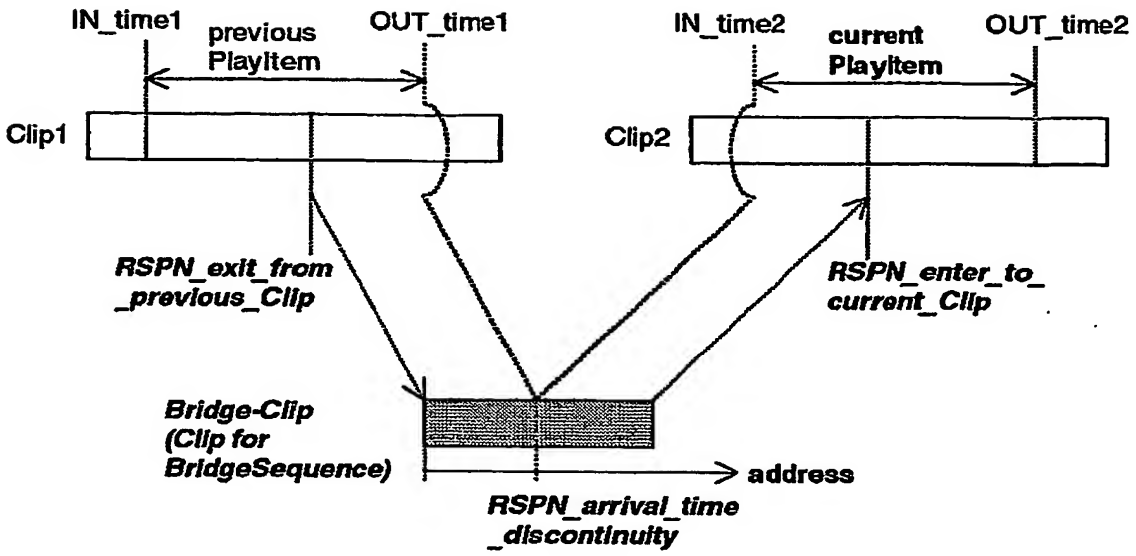


(D) connection_condition='10'

connection_condition='11'

connection_condition の説明

【図３７】

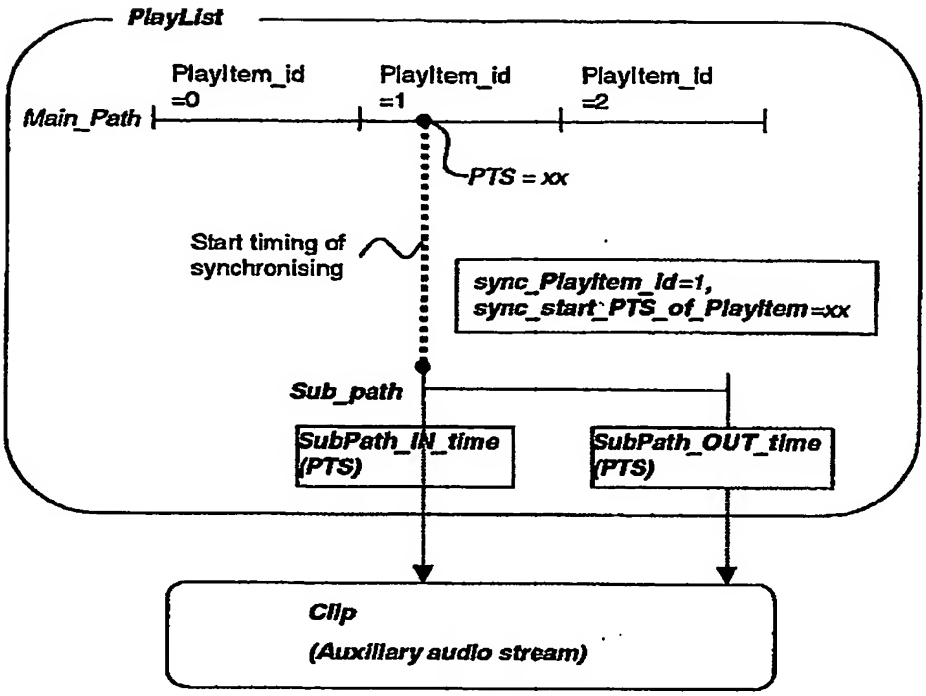

IN_time1
previous
PlayItem
OUT_time1
IN_time2
current
PlayItem
OUT_time2
Clip1
Clip2
RSPN_exit_from
_previous_Clip
RSPN_enter_to_
current_Clip
Bridge-Clip
(Clip for
BridgeSequence)
address
RSPN_arrival_time
_discontinuity

【図３８】

Syntax	No. of bits	Mnemonics
BridgeSequenceInfo() {		
Bridge_Clip_Information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

BridgeSequenceInfoのシンタクス

【図３９】


PlayList
PlayItem_id =0
PlayItem_id =1
PlayItem_id =2
Main_Path
PTS = xx
Start timing of synchronising
sync_PlayItem_id=1,
sync_start_PTS_of_PlayItem=xx
Sub_path
SubPath_IN_time (PTS)
SubPath_OUT_time (PTS)
Clip
(Auxiliary audio stream)

【図４０】

Syntax	No. of bits	Mnemonics
SubPlayItem() {		
Clip_Information_file_name	8*10	bslbf
SubPath_type	8	bslbf
sync_PlayItem_id	8	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
}		

SubPlayItemのシンタクス

【図４１】

SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

【図４２】

Syntax	No. of bits	Mnemonics
PlayListMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for(i=0; i < *number_of_PlayList_marks*; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
PlayItem_id	8	uimsbf
reserved	24	uimsbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

PlayListMarkのシンタクス

【図４３】

Mark_type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、０または１でなければならない。
0x01	book-mark	PlayList の再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark() において定義されるスキップマークポイントの数は、０または１でなければならない。
0x03 - 0x8F	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図４４】

CPI_type in the PlayList()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、*TU_map_time_axis* 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さねばならない。mark_time_stamp は、次に示す等式により計算される。 mark_time_stamp = *TU_start_time* % 2^{32}

mark_time_stamp

【図４５】

Syntax	No. of bits	Mnemonics
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakerPrivateData_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
STC_Info()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0; i<N3; i++){		
padding_word	16	bslbf
}		
CPI()		
for(i=0; i<N4; i++){		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0; i<N5; i++){		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

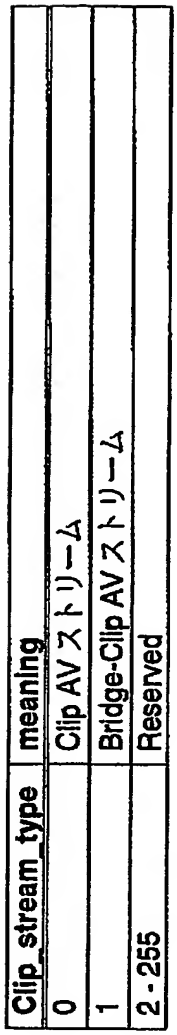
zzzzz.clpi のシンタクス

【図４６】

Syntax	No. of bits	Mnemonics
ClipInfo() {		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_servece_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
servece_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_future_use	256	bslbf
}		

ClipInfoのシンタクス

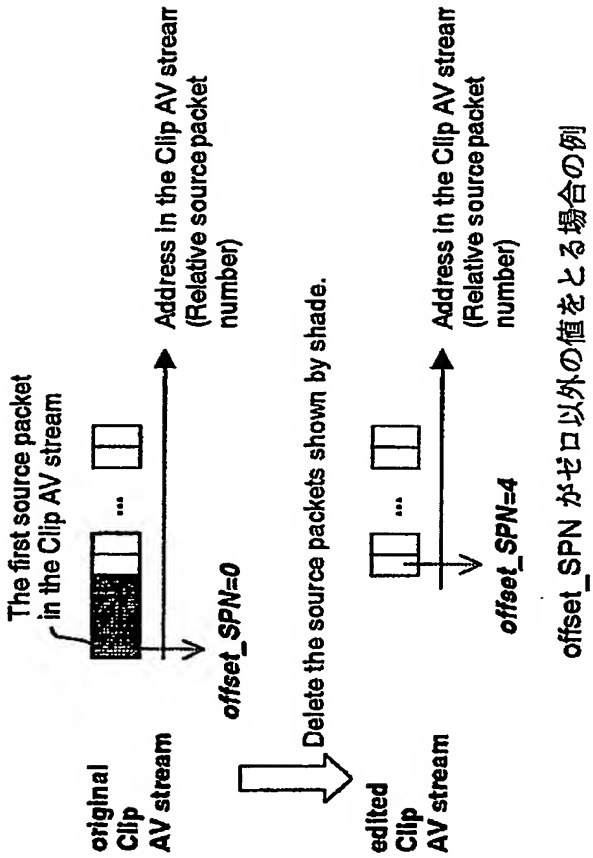
【図４７】



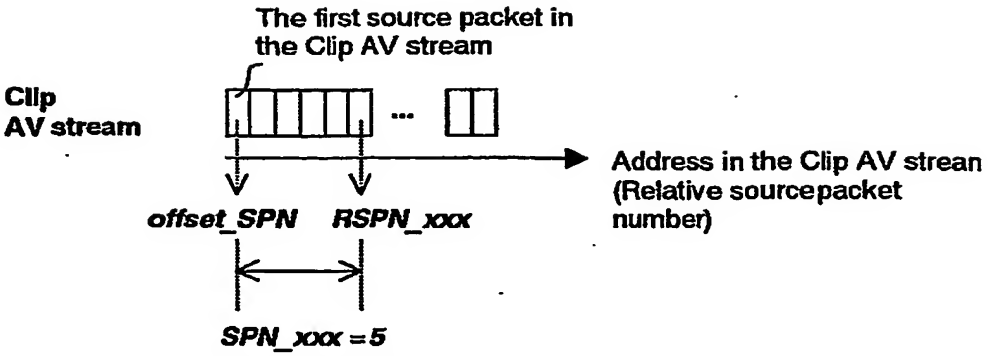
Clip_stream_type	meaning
0	Clip AVストリーム
1	Bridge-Clip AVストリーム
2 - 255	Reserved

Clip_stream_type

【図４８】

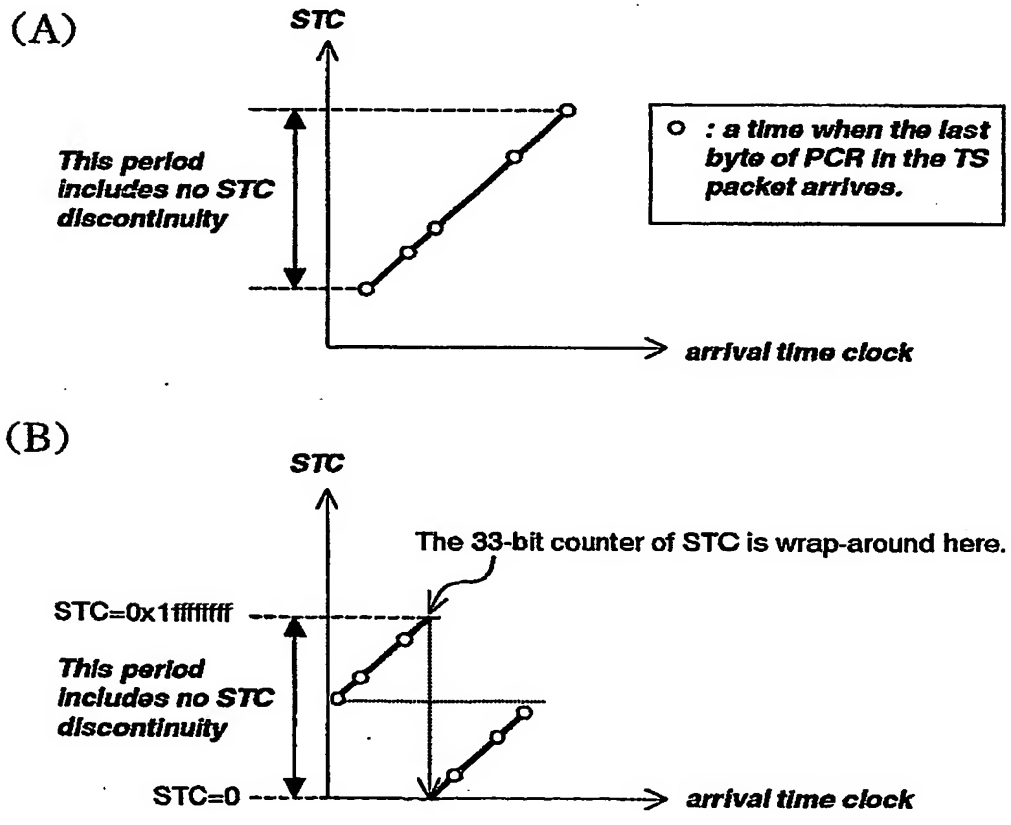

The first source packet in the Clip AV stream
original Clip AV stream
offset_SPN=0
Address in the Clip AV stream (Relative source packet number)
Delete the source packets shown by shade.
edited Clip AV stream
offset_SPN=4
Address in the Clip AV stream (Relative source packet number)
offset_SPN がゼロ以外の値をとる場合の例

【図４９】

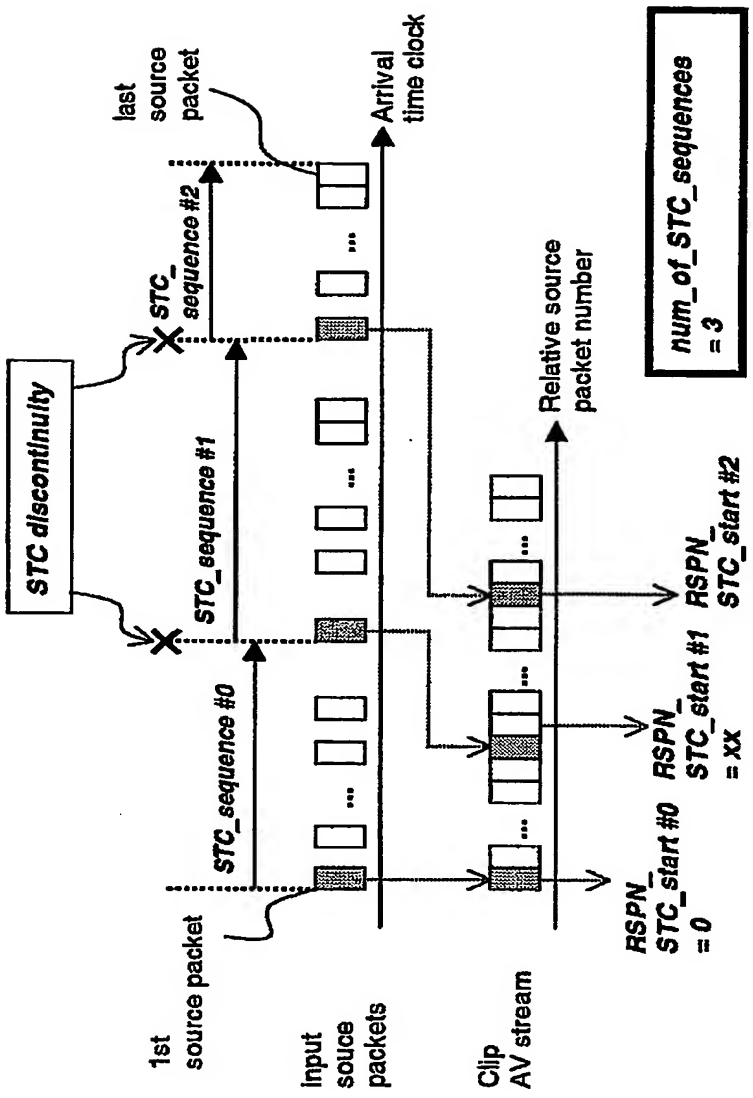


AVストリームでの offset_SPN と相対ソースパケット番号（RSPN_xxx）の間の関係

【図５０】



【図５１】


last source packet
Arrival time clock
num_of_STC_sequences = 3
STC_ sequence #2
STC discontinuity
STC_sequence #1
Relative source packet number
STC_sequence #0
RSPN_ STC_start #2 = yy
RSPN_ STC_start #1 = xx
RSPN_ STC_start #0 = 0
1st source packet
Input souce packets
Clip AV stream
--STC_Info

【図５２】

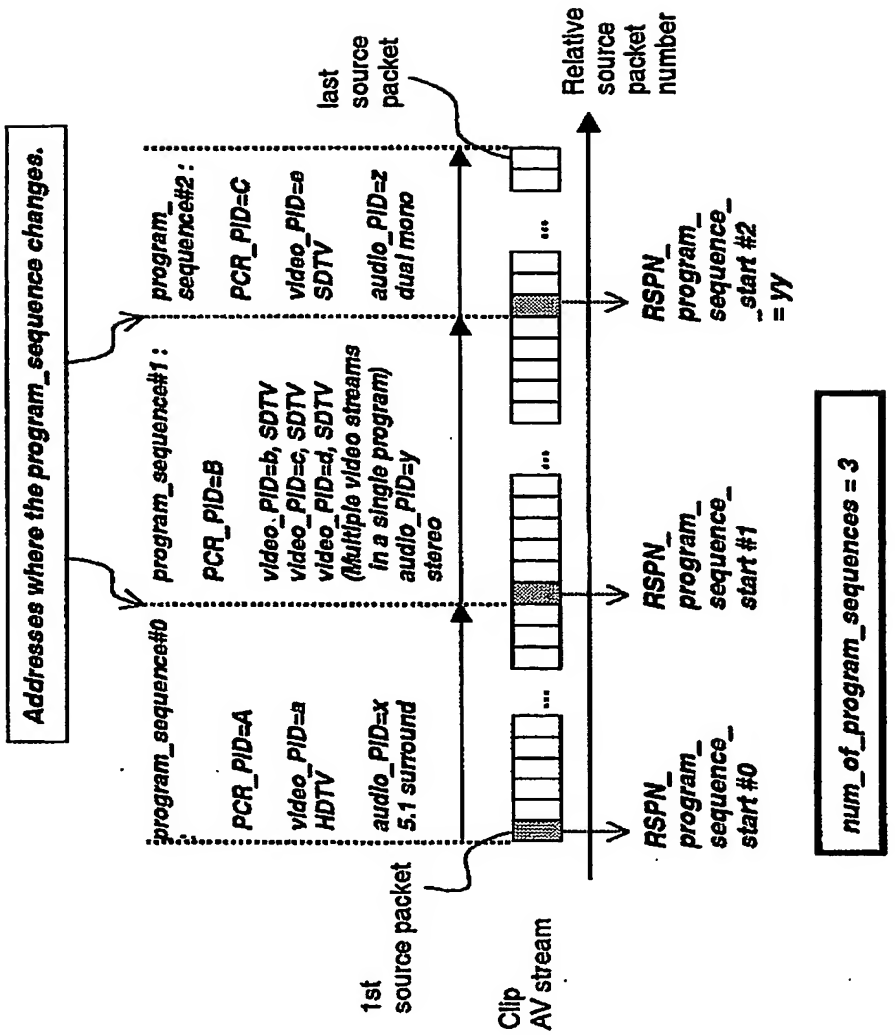
Syntax	No. of bits	Mnemonics
STC_Info() {		
version_number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
num_of_STC_sequences	8	uimsbf

for(STC_sequence_id=0;
STC_sequence_id < num_of_STC_sequences;
STC_sequence_id++) {		

reserved	32	bslbf
RSPN_STC_start	32	uimsbf
}		
}		
}		

STC_Infoのシンタクス

【図５３】



ProgramInfo の例

Syntax	No. of bits	Mnemonics
ProgramInfo() {		
version_number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
number_of_program_sequences	8	uimsbf
for(i=0; i<*number_of_program_sequences*; i++){		
RSPN_program_sequence_start	32	uimsbf
reserved	48	bslbf
PCR_PID	16	bslbf
number_of_videos	8	uimsbf
number_of_audios	8	uimsbf
for (k=0; k<*number_of_videos*; k++) {		
video_stream_PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0; k<*number_of_audios*; k++) {		
audio_stream_PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		
}		

ProgramInfoのシンタクス

出証特２００１－３０１４４２９

【図５５】

Syntax	No. of bits	Mnemonics
VideoCodingInfo() {		
video_format	8	uimsbf
frame_rate	8	uimsbf
display_aspect_ratio	8	uimsbf
reserved	8	bslbf
}		

VideoCodingInfoのシンタクス

【図５６】

video_format	Meaning
0	480i
1	576i
2	480p (including 640x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

vidoe_format

【図５７】

frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94 ..)
8	60
9 - 254	reserved
255	No information

frame_rate

【図５８】

display_aspect_ratio	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

display_aspect_ratio

【図５９】

Syntax	No. of bits	Mnemonics
AudioCodingInfo() {		
audio_coding	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bslbf
}		

AudioCodingInfoのシンタクス

【図６０】

audio_coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

audio_coding

【図６１】

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

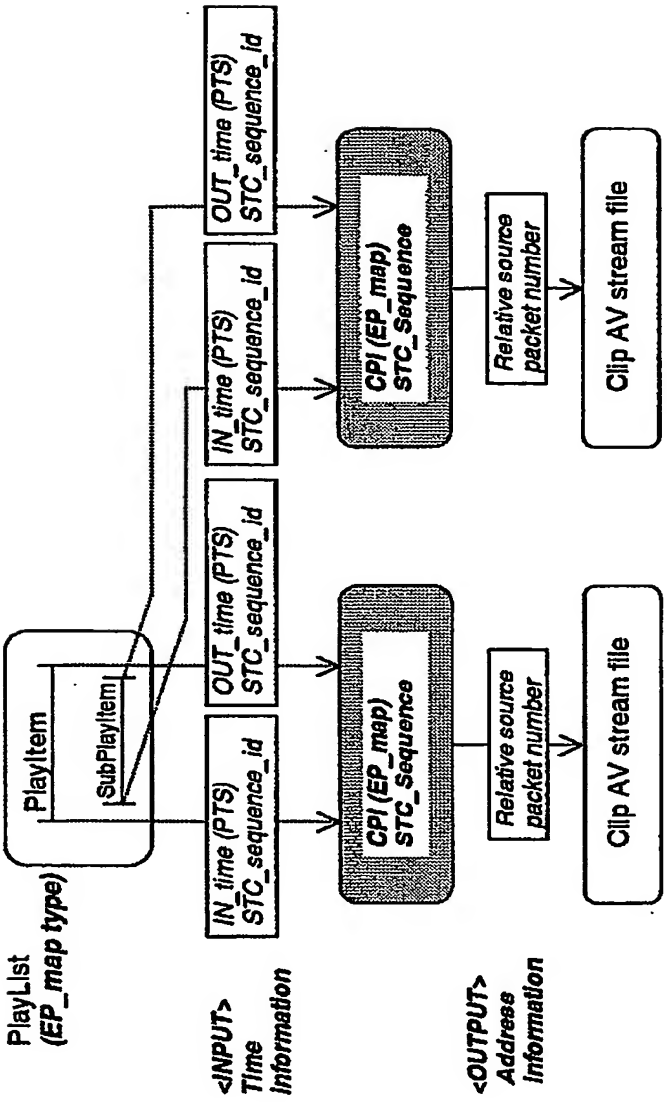
audio_component_type

【図６２】

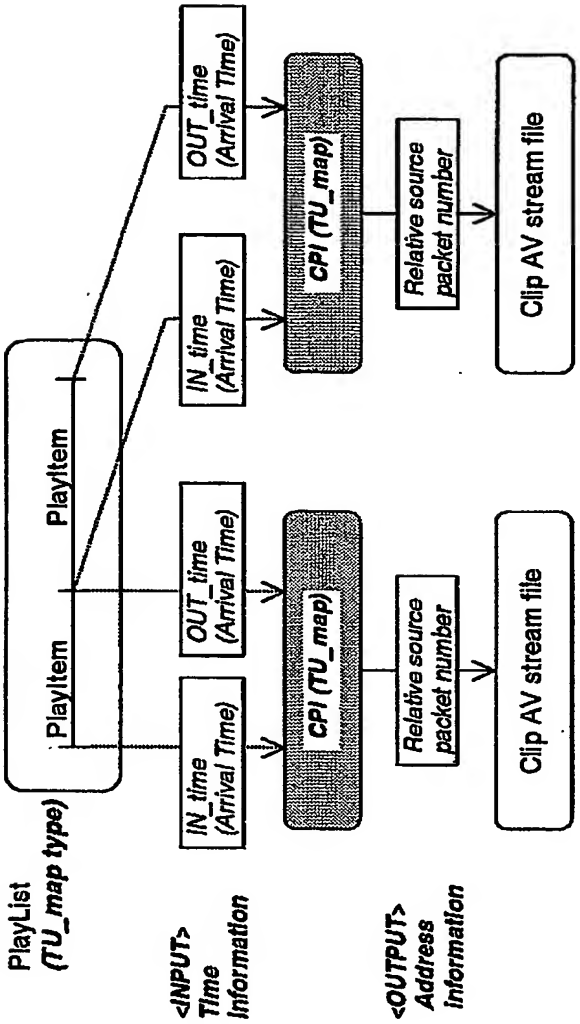
sampling_frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【図６３】


PlayList
(EP_map type)
PlayItem
SubPlayItem
<INPUT>
Time
Information
IN_time (PTS)
STC_sequence_id
OUT_time (PTS)
STC_sequence_id
IN_time (PTS)
STC_sequence_id
OUT_time (PTS)
STC_sequence_id
<OUTPUT>
Address
Information
CPI (EP_map)
STC_Sequence
CPI (EP_map)
STC_Sequence
Relative source
packet number
Relative source
packet number
Clip AV stream file
Clip AV stream file

【図６４】


PlayList
(TU_map type)
PlayItem
PlayItem
<INPUT>
Time
Information
IN_time
(Arrival Time)
OUT_time
(Arrival Time)
IN_time
(Arrival Time)
OUT_time
(Arrival Time)
CPI (TU_map)
CPI (TU_map)
<OUTPUT>
Address
Information
Relative source
packet number
Relative source
packet number
Clip AV stream file
Clip AV stream file

【図６５】

Syntax	No. of bits	Mnemonics
CPI() {		
version_number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI_type	1	bslbf
if (CPI_type == 0)		
EP_map()		
else		
TU_map()		
}		

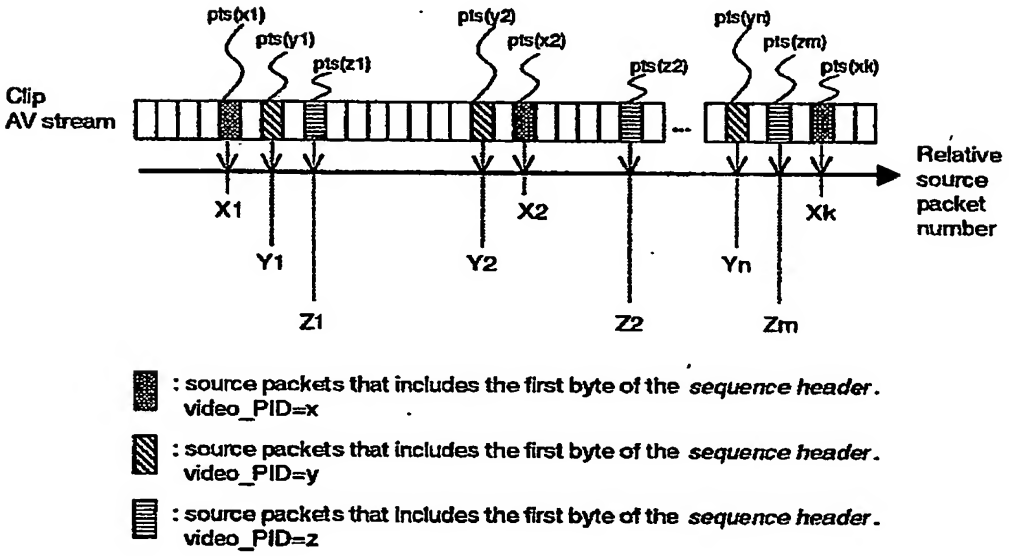
CPIのシンタクス

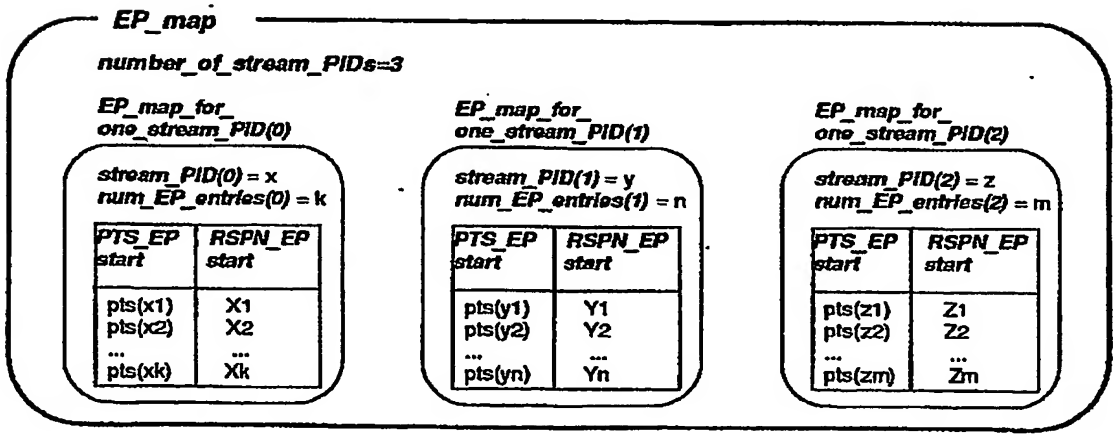
【図６６】

CPI_type	Meaning
0	EP_map type
1	TU_map type

CPI_type の意味

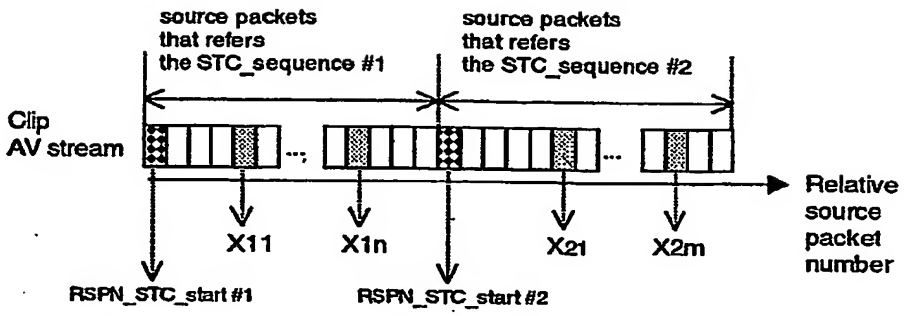
【図６７】





ビデオの EP_map の例

【図６８】



: source packets that includes the first byte of the *sequence header.* video_PID=x

: source packets that referred by *RSPN_STC_start (defined in the STC_Info)*

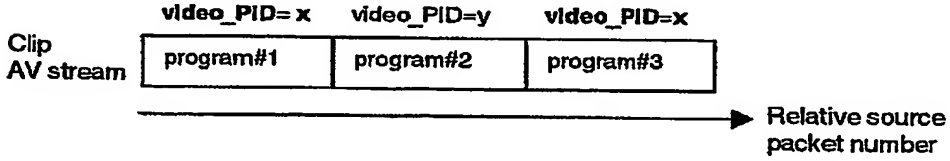
EP_map_for_one_stream_PID

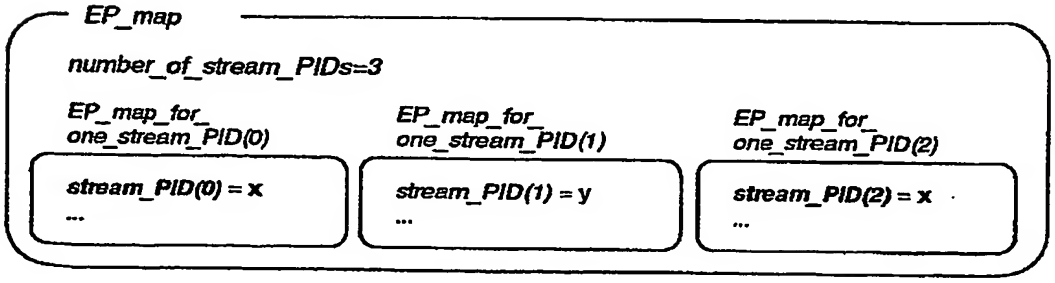
video_PID=x

PTS_EP start	*RSPN_EP start*	
pts(x11)	X11	*These data belong to the STC_sequence #1*
...	...	
pts(x1n)	X1n	
		boundary
pts(x21)	X21	*These data belong to the STC_sequence #2*
...	...	
pts(x2m)	X2m	

RSPN_STC_start #2 < X21

【図６９】





【図７０】

Syntax	Bits	Mnemonics
Thumbnail() {		
version_number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for(l = 0; l < *number of thumbnails*; l++) {		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k = 0; k <*number of tn_blocks*; k++) {		
tn_block	tn_block_size* 1024*8	
}		
}		
}		

Thumbnailのシンタクス

【図７１】

EP_type	Meaning
0	video
1	audio
2 - 15	reserved

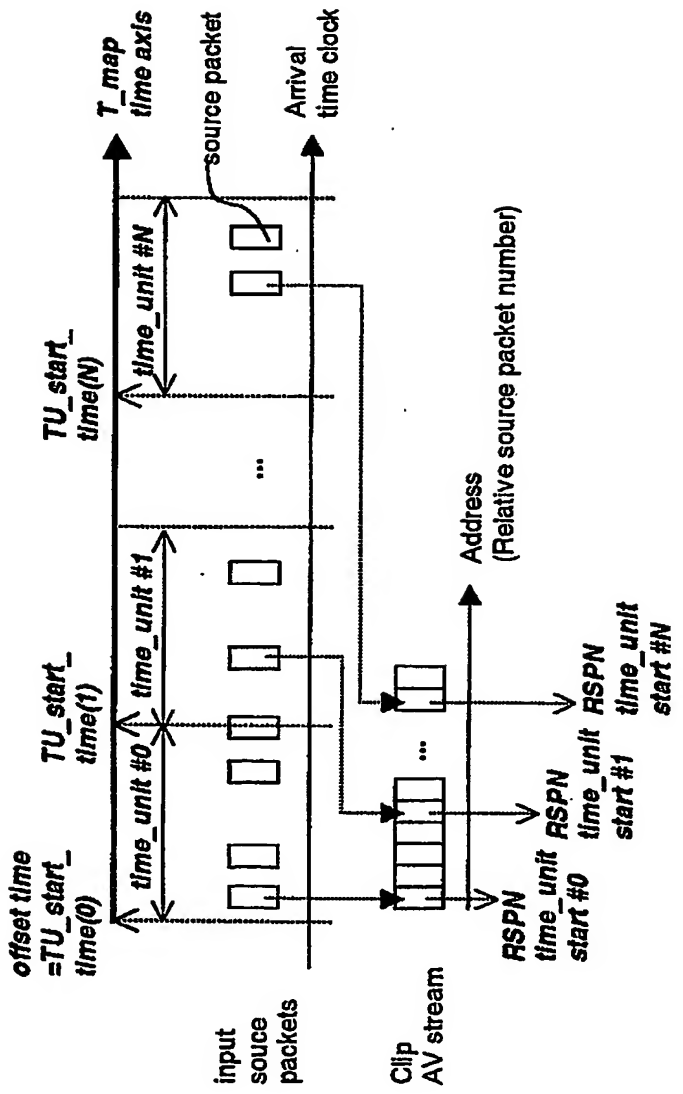
EP_type Values

【図７２】

Syntax	No. of bits	Mnemonics
EP_map_for_one_stream_PID(N){		
for (i=0; i< N; i++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

EP_map_for_one_stream_PID のシンタクス

【図７３】


T_map
time axis
source packet
Arrival
time clock
TU_start_
time(N)
time_unit #N
TU_start_
time(1)
time_unit #1
time_unit #0
offset time
=TU_start_
time(0)
Address
(Relative source packet number)
RSPN
time_unit
start #0
RSPN
time_unit
start #1
RSPN
time_unit
start #N
input
souce
packets
Clip
AV stream

【図７４】

Syntax	No. of bits	Mnemonics
TU_map0{		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0; k<*number_of_time_unit_entries*; k++)		
RSPN_time_unit_start	32	uimsbf
}		

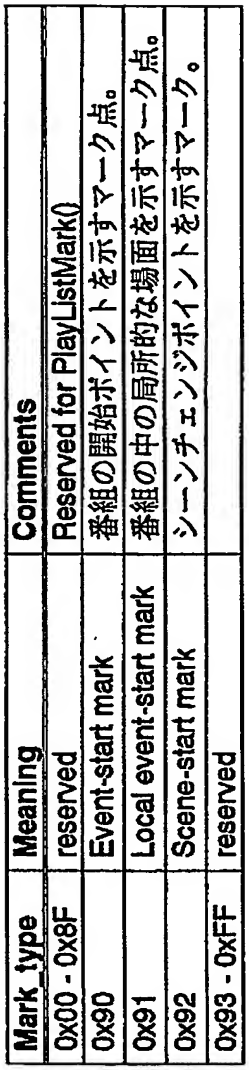
TU_mapのシンタクス

【図７５】

Syntax	No. of bits	Mnemonics
ClipMark() {		
version_number	8*4	bslbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for(i=0; i < *number_of_Clip_marks*; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bslbf
character_set	8	bslbf
name_length	8	uimsbf
mark_name	8*256	bslbf
ref_thumbnail_index	16	uimsbf
}		
}		

ClipMarkのシンタクス

【図７６】



Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーンチェンジポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

【図７７】

CPI_type in the CPI()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、*TU_map_time_axis* 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表されねばならない。mark_time_stamp は、次に示す等式により計算される。 mark_time_stamp = *TU_start_time* % 2^{32}

mark_type_stamp

【図７８】

Syntax	No. of bits	Mnemonics
menu.thmb / mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for(i=0; i<N1; i++)		
padding_word	16	bslbf
}		

menu.thmb と mark.thmb のシンタクス

Syntax	Bits	Mnemonics
Thumbnail() {		
version_number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for(i = 0; i < *number_of_thumbnails*; i++) {		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	8*2*L1	bslbf
for(k = 0; k <*number_of_tn_blocks*; k++) {		
tn_block	tn_block_size* 1024*8	
}		
}		
}		

Thumbnailのシンタクス

出証特２００１－３０１４４２９

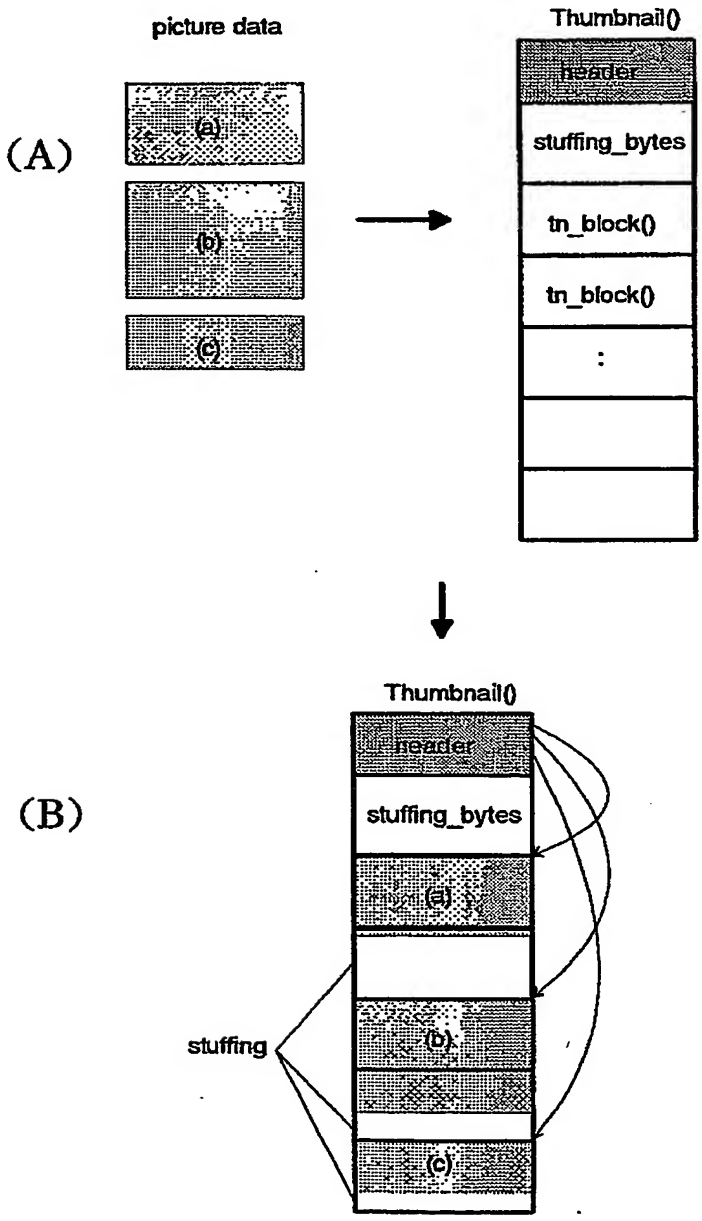
特２０００－１８３７７０

【図８０】

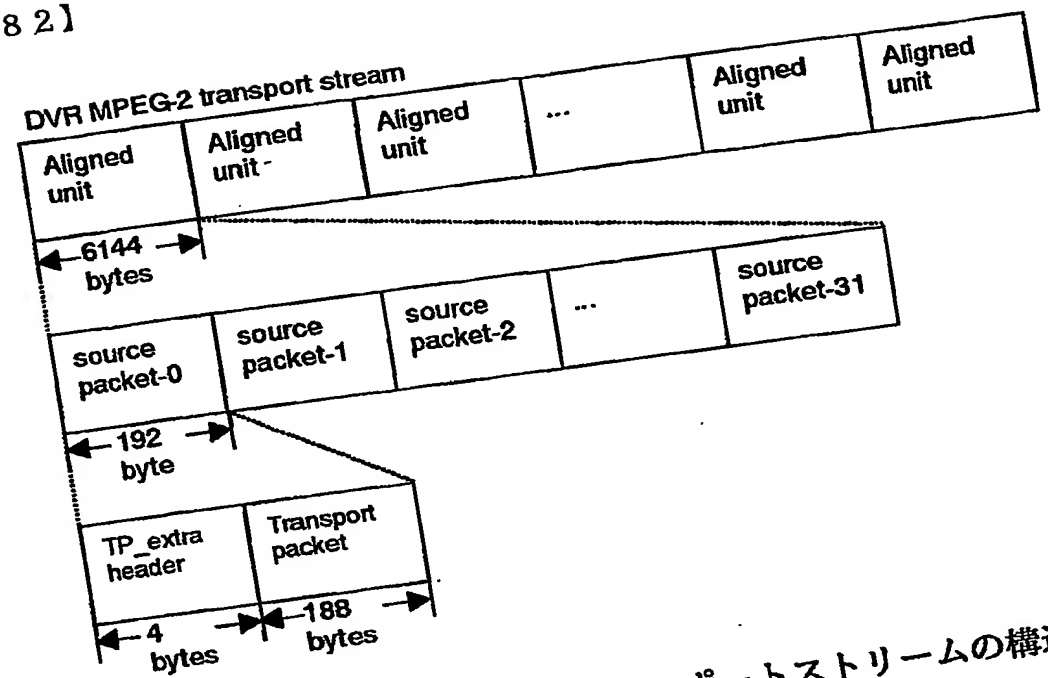
Thumbnail_picture_format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

thumbnail__picture__format

【図８１】

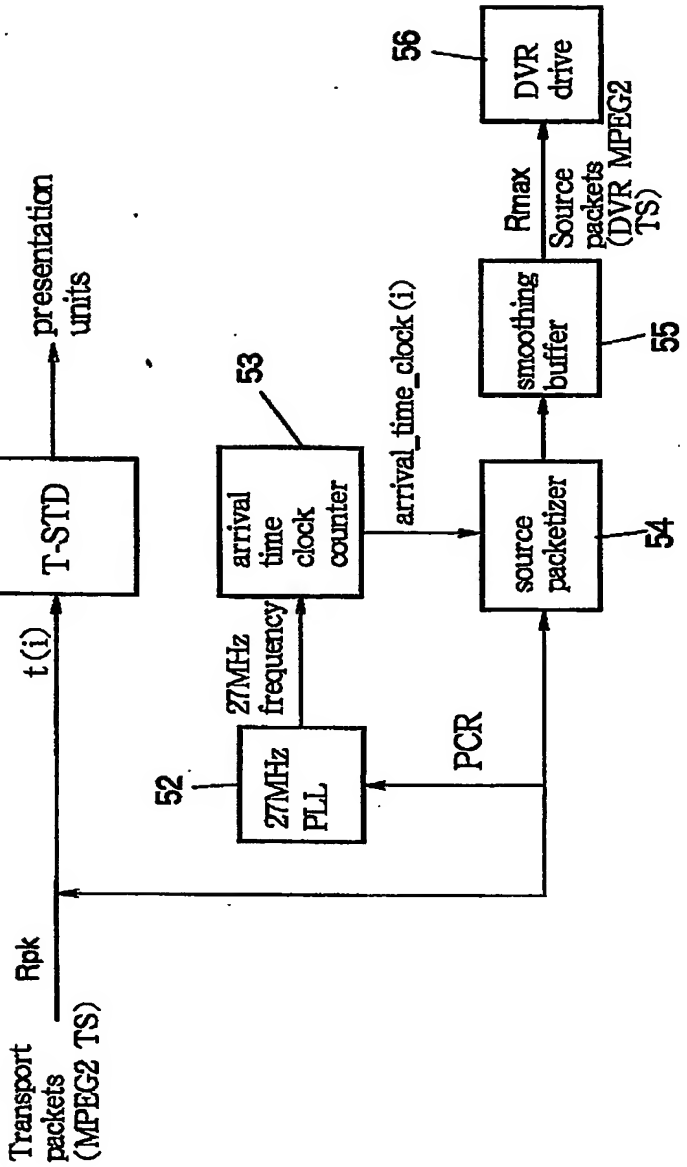


【図８２】



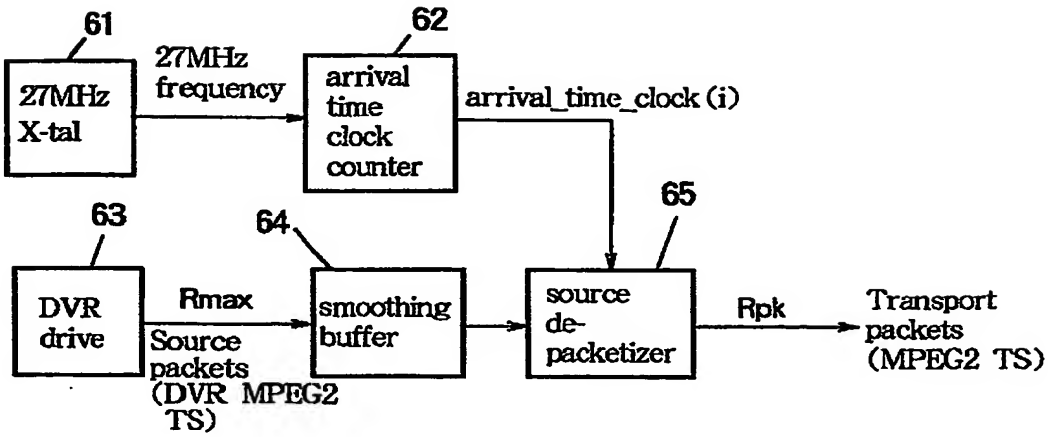
DVR MPEG-2　トランスポートストリームの構造

【図８３】



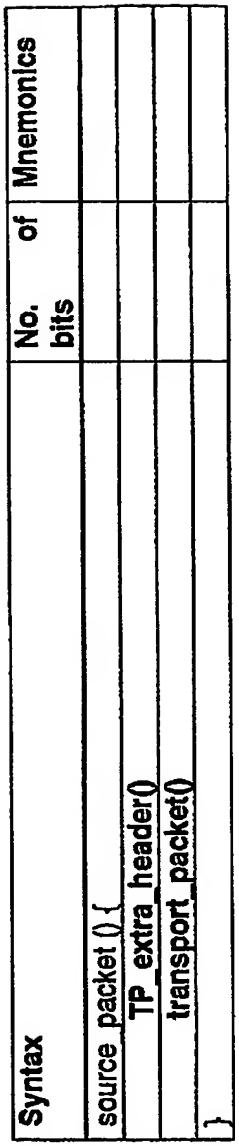
DVR MPEG-2 トランスポートストリームのレコーダモデル

【図８４】



DVR MPEG-2　トランスポートストリームのプレーヤモデル

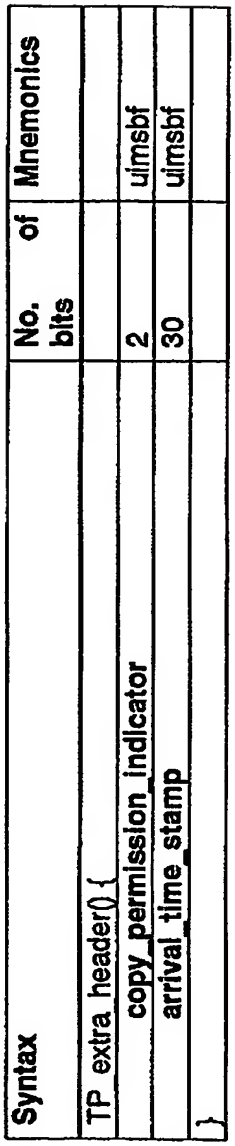
【図８５】



Syntax	No. of bits	Mnemonics
source_packet () {		
TP_extra_header()		
transport_packet()		
}		

source packet

【図８６】



Syntax	No. of bits	Mnemonics
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

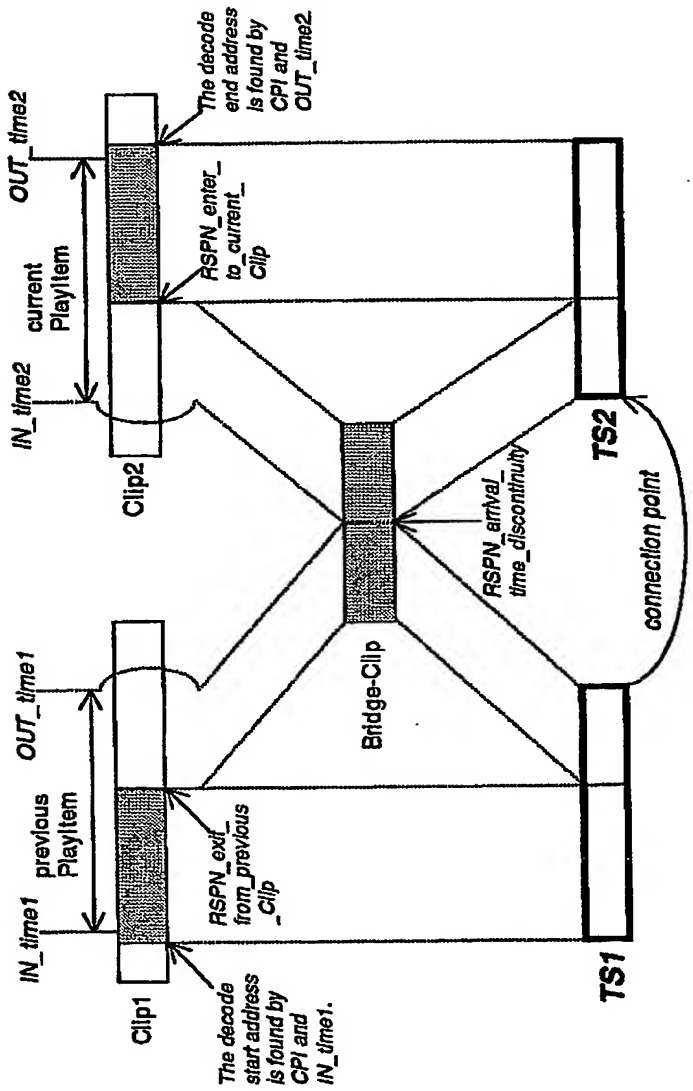
TP_extra_header

【図８７】

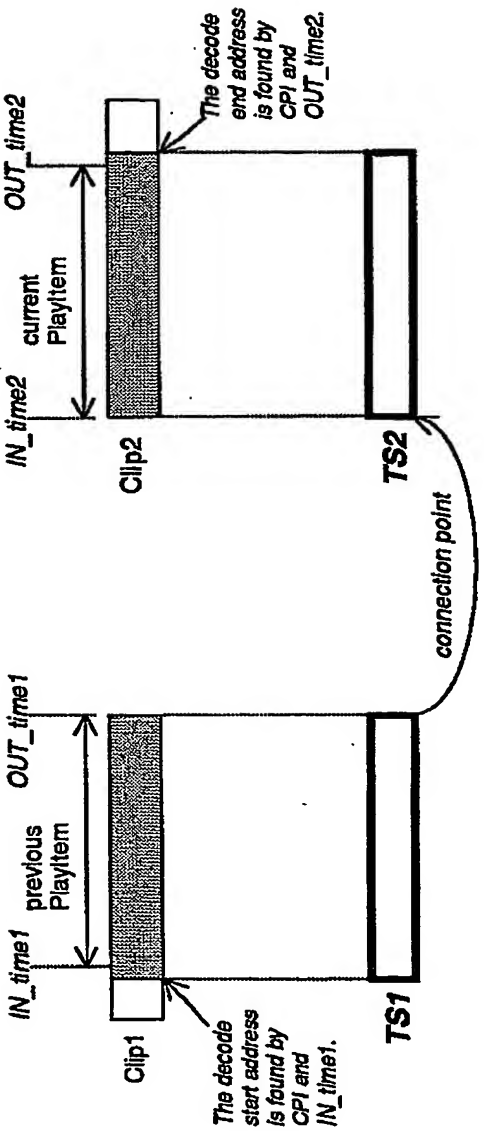
copy_permission_indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

· copy permission indicator table

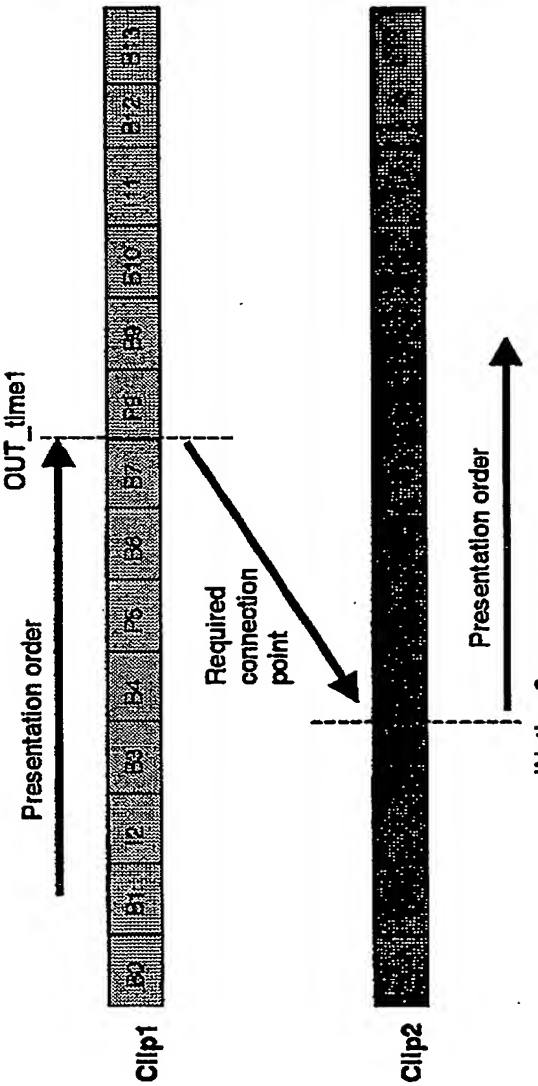
【図８８】


The decode end address is found by CPI and OUT_time2
OUT_time2
RSPN_enter_ to_current_ Clip
current PlayItem
IN_time2
Clip2
Bridge-Clip
RSPN_arrival_ time_discontinuity
TS2
connection point
OUT_time1
previous PlayItem
RSPN_exit_ from_previous _Clip
IN_time1
Clip1
The decode start address is found by CPI and IN_time1.
TS1

【図８９】

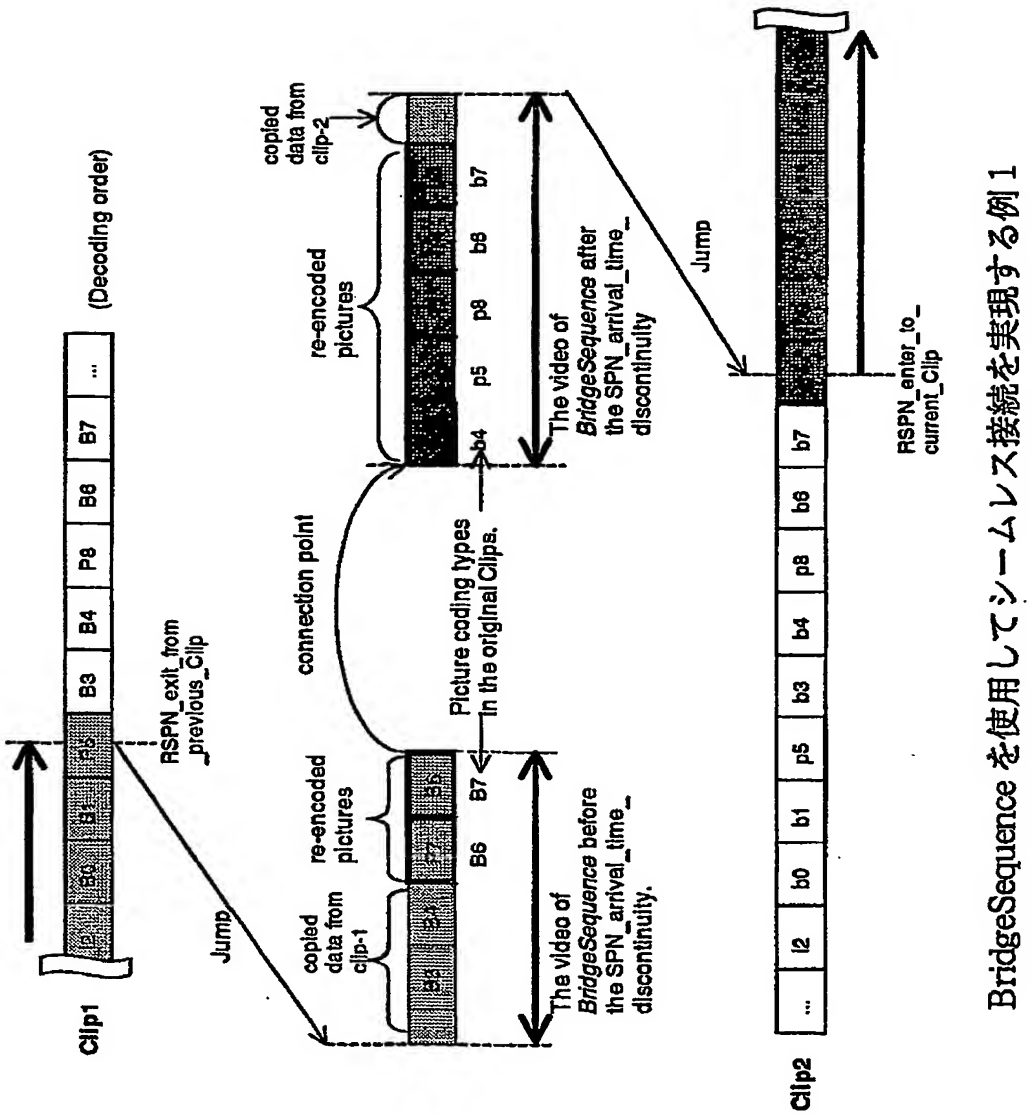

The decode end address is found by CPI and OUT_time2.
OUT_time2
current PlayItem
IN_time2
Clip2
TS2
connection point
OUT_time1
previous PlayItem
IN_time1
Clip1
TS1
The decode start address is found by CPI and IN_time1.

【図９０】



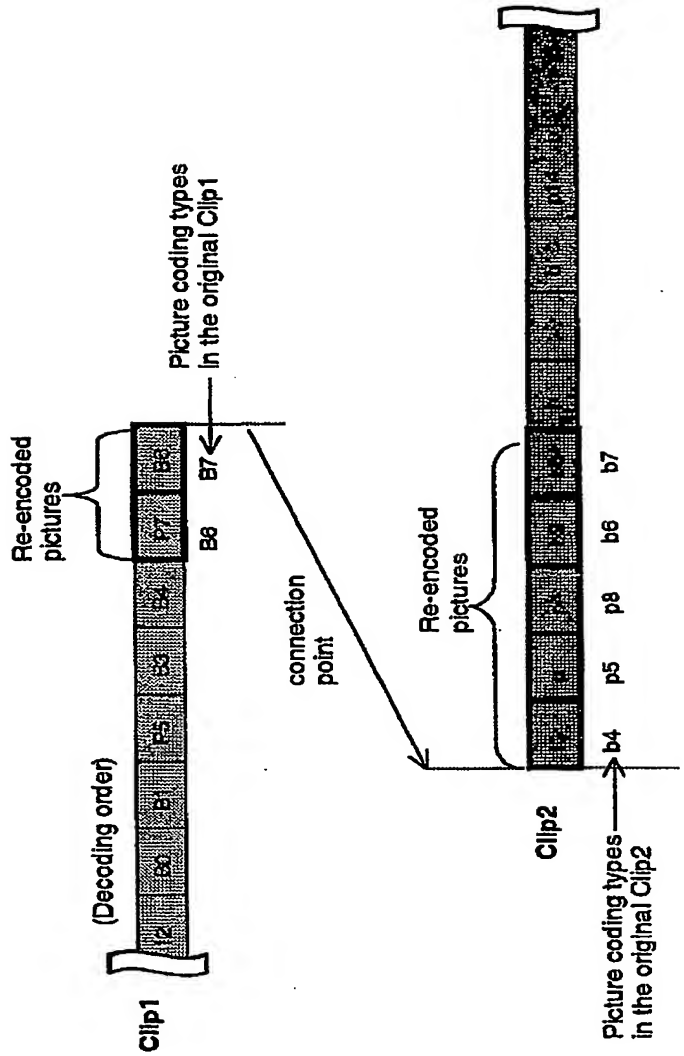
ピクチャの表示順序で示すシームレス接続の例

【図９１】



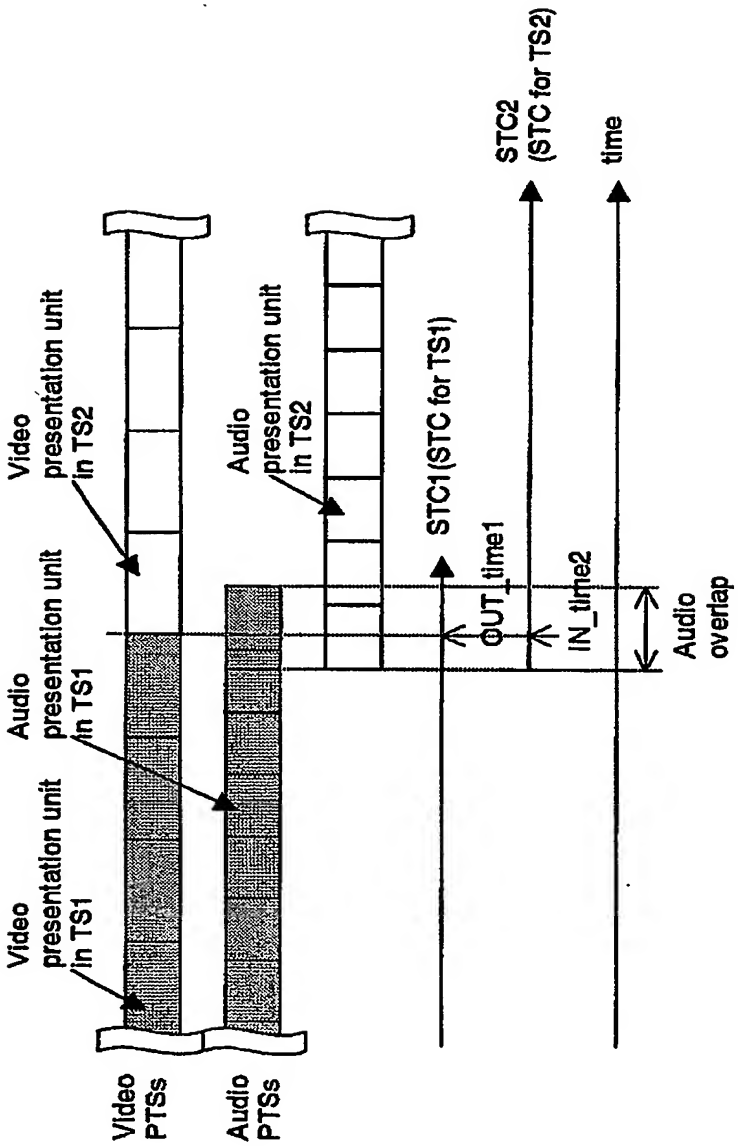
BridgeSequence を使用してシームレス接続を実現する例１

【図９２】

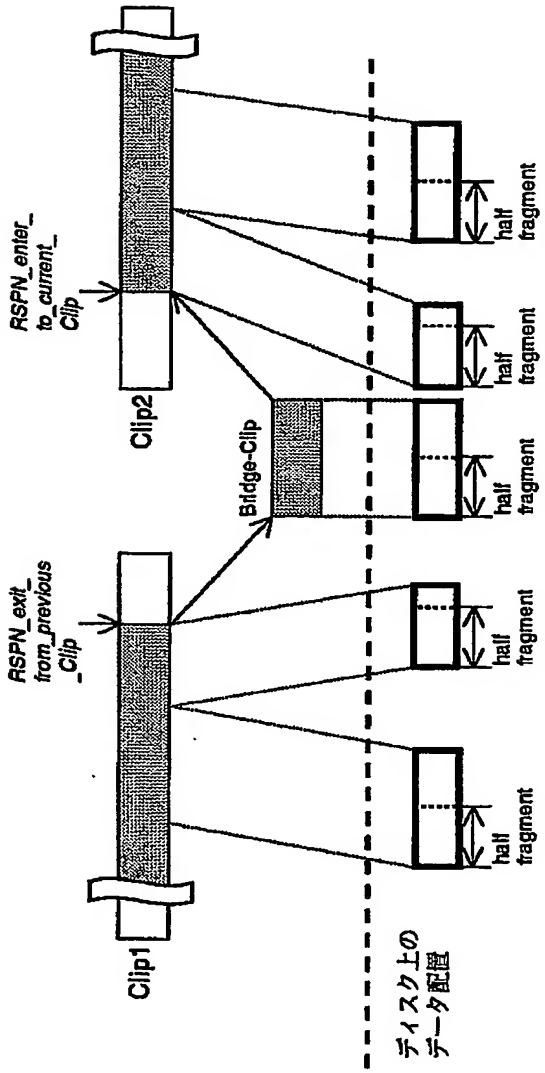


BridgeSequence を使用しないでシームレス接続を実現する例２

【図９３】

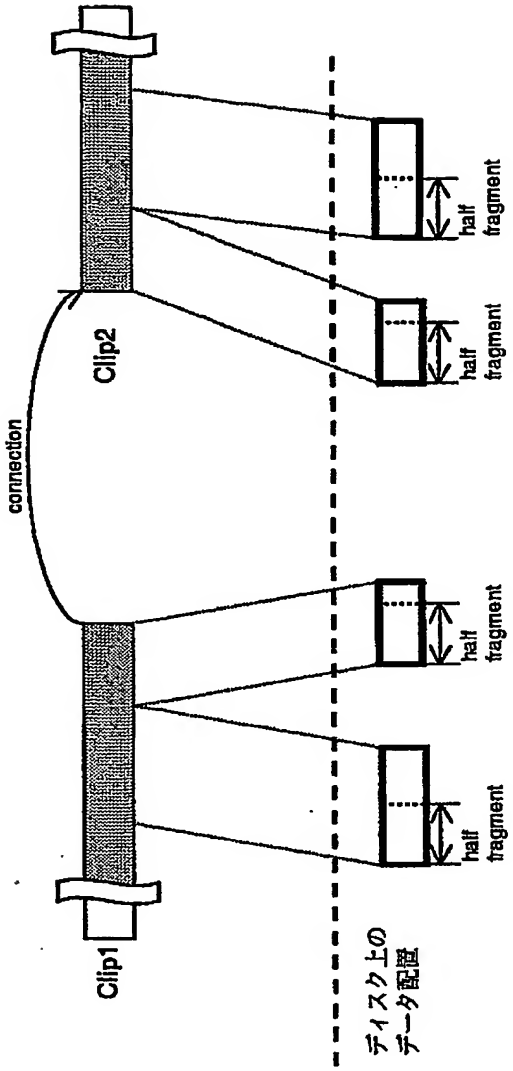

Video presentation unit in TS1
Audio presentation unit in TS1
Video presentation unit in TS2
Audio presentation unit in TS2
Video PTSs
Audio PTSs
STC1(STC for TS1)
STC2 (STC for TS2)
time
OUT_time1
IN_time2
Audio overlap

【図９４】



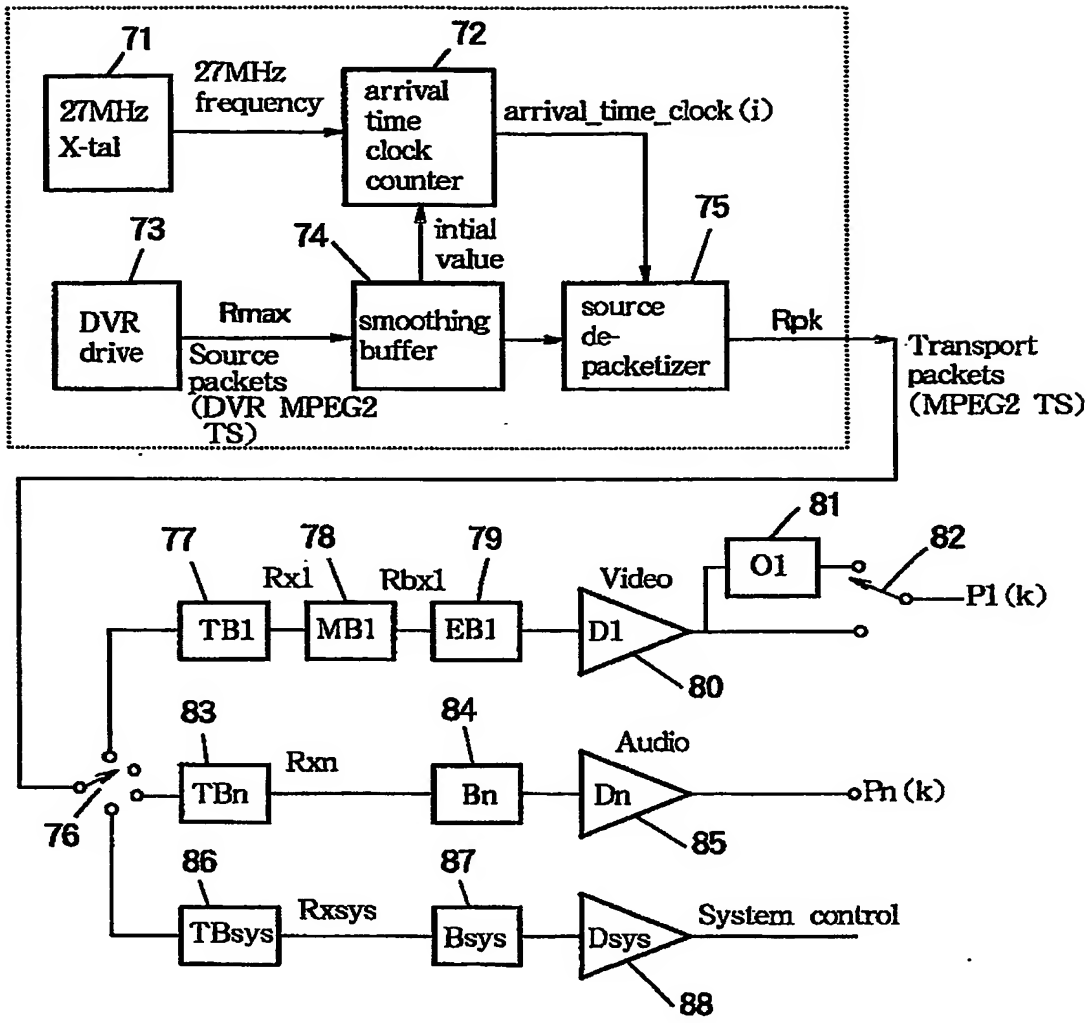
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図９５】

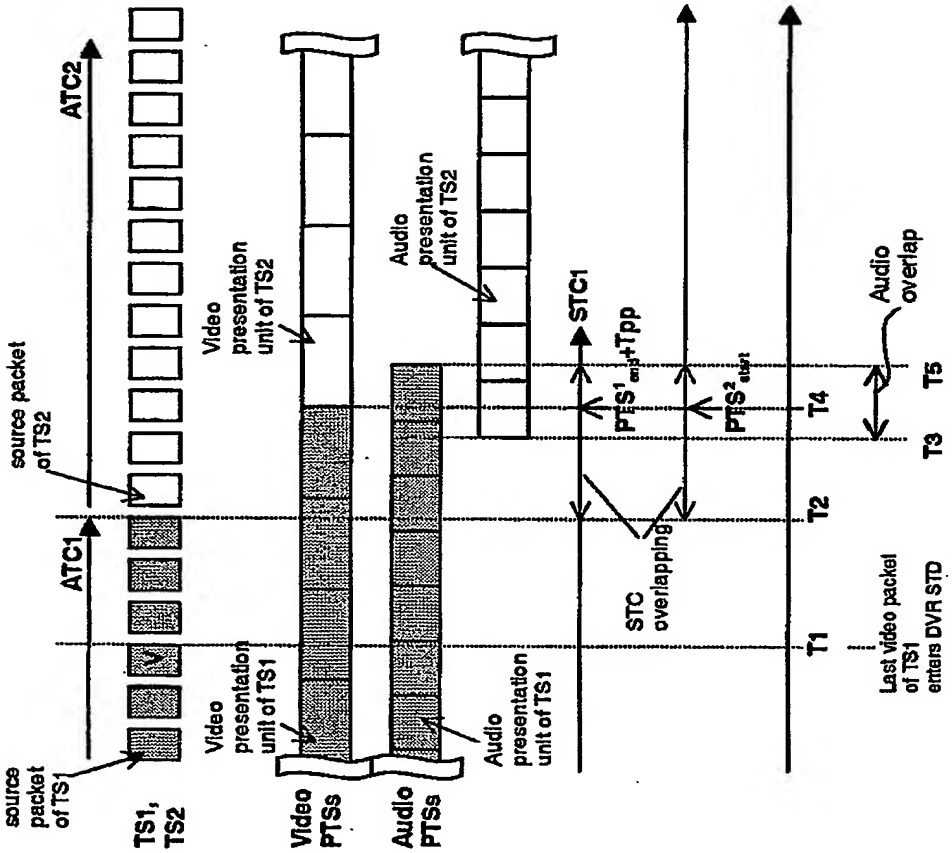


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

【図９６】

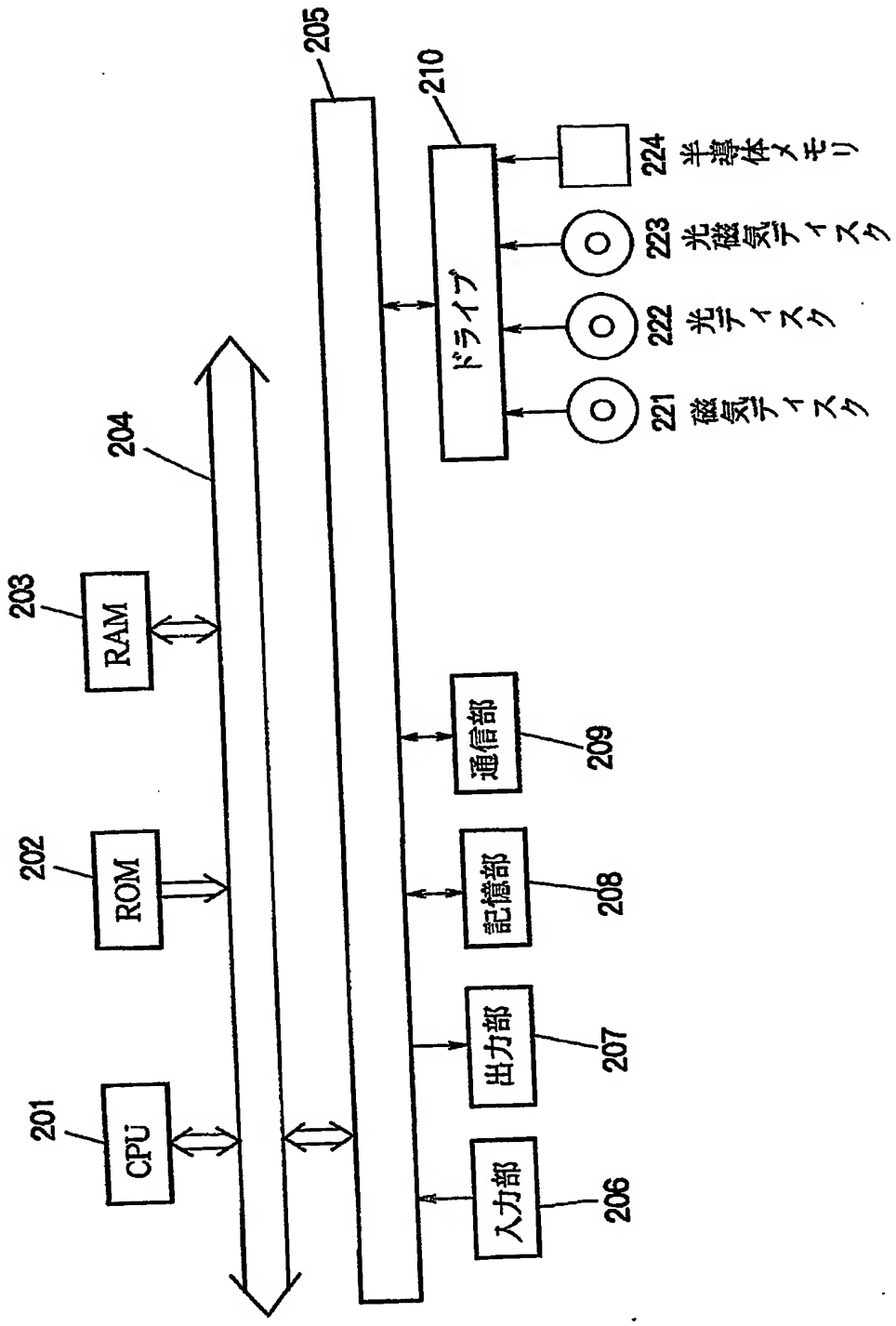

71
27MHz X-tal
27MHz frequency
72
arrival time clock counter
arrival_time_clock (i)
intial value
73
DVR drive
74
Rmax
Source packets (DVR MPEG2 TS)
smoothing buffer
75
source de-packetizer
Rpk
Transport packets (MPEG2 TS)
77
TB1
78
Rx1
MB1
Rbx1
79
EB1
Video
D1
80
81
O1
82
P1 (k)
83
TBn
Rxn
84
Bn
Audio
Dn
85
Pn (k)
76
86
TBsys
Rxsys
87
Bsys
Dsys
System control
88

【図９７】



あるAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

【図９８】


205
210
224
半導体メモリ
223
光磁気ディスク
222
光ディスク
221
磁気ディスク
ドライブ
204
203
RAM
202
ROM
201
CPU
通信部
209
記憶部
208
出力部
207
入力部
206

【書類名】　　要約書

【要約】

【課題】　AVストリームの読み出し位置の決定や復号処理を速やかに行えるようにする。

【解決手段】　ClipInfoは、AVストリームの属性情報を蓄積する。ClipInfo内のOffset_SPNは、AVストリームの最初のソースパケットについてのソースパケット番号のオフセット値を与える。time_controlled_flagは、AVストリームが時間経過に対してファイルサイズが比例するように記録されている記録モードであるか否かを示す。その他のデータは、AVストリームのタイプを示すデータや、記録された日時に関するデータである。これらのデータを参照することにより、AVストリームの読み出し位置の決定を行うことができる。

【選択図】　　図４６

出願人履歴情報

識別番号　〔０００００２１８５〕

１．変更年月日　１９９０年　８月３０日
［変更理由］　新規登録
住　所　東京都品川区北品川６丁目７番３５号
氏　名　ソニー株式会社

出証特２００１－３０１４４２９